Assume-Admissible Synthesis *

Romain Brenguier
Department of Computer Science, University of Oxford, UK

Jean-François Raskin
Département d'informatique, Université Libre de Bruxelles, Belgium

Ocan Sankur
CNRS, Irisa, France

September 28, 2016

Abstract

In this paper, we introduce a novel rule for synthesis of reactive systems, applicable to systems made of n components which have each their own objectives. This rule is based on the notion of *admissible* strategies. We compare this rule with previous rules defined in the literature, and show that contrary to the previous proposals, it defines sets of solutions which are *rectangular*. This property leads to solutions which are robust and resilient, and allows one to synthesize strategies separately for each agent. We provide algorithms with optimal complexity and also an abstraction framework compatible with the new rule.

1 Introduction

The automatic synthesis of reactive systems has recently attracted a considerable attention. The theoretical foundations of most of the contributions in this area rely on two-player zero sum games played on graphs: one player (player 1) models the system to synthesize, and the other player (player 2) models its environment. The game is zero-sum: the objective of player 1 is to enforce the specification of the system while the objective of player 2 is the *negation* of this specification. This is a *worst-case* assumption: because the cooperation of the environment cannot be assumed, we postulate that it is *antagonistic*.

A fully adversarial environment is usually a bold abstraction of reality. Nevertheless, it is popular because it is *simple* and *sound*: a *winning strategy* against an antagonistic player is winning against *any* environment which pursues its own objective. But this approach may fail to find a winning strategy even if there exist solutions when the objective of the environment is taken into account. Also, this model is for two players only: system *vs* environment. In practice, both the system and the environment may be composed of several parts to be constructed individually or whose objectives should be considered one at a time. In fact, many systems, such as telecommunication protocols, and distributed algorithms are made of several components or processes, each having its own objective which may or may not conflict other components' objectives. Consider, for instance, a communication network in which each node has the objective of transmitting a message to a subset of other nodes, using some preferred frequency range;

*Supported by the ERC starting grant inVEST (FP7-279499)

the objectives of some nodes may not conflict at all if they are independent (using different frequencies), while some of them may be in conflict. Indeed, game theory is used to model such situations; see *e.g.* [20]. Such problems are the subject of *non-zero sum games* where each entitiy having its own objective is seen as a different *player* (a.k.a. *agent*). For controller synthesis within this context, it is thus crucial to take different players' objectives into account when synthesizing strategies; accordingly, alternative notions have been proposed in the literature.

A first classical alternative is to *weaken* the winning condition of player 1 using the objective of the environment, requiring the system to win only when the environment meets its objective. This approach together with its weaknesses have been discussed in [4], we will add to that later in the paper. A second alternative is to use concepts from n-players *non*-zero sum games. This is the approach taken both by *assume-guarantee synthesis* [8] (AG), and by *rational synthesis* [18] (RS). For two players, AG relies on *secure equilibria* [10] (SE), a refinement of Nash equilibria [28] (NE). In SE, objectives are lexicographic: players first try to maximize their own specifications, and then try to falsify the specifications of others. It is shown in [10] that SE are those NE which represent enforceable contracts between the two players. However the AG rule as extended to several players in [8] no longer corresponds to secure equilibria. This was not noticed in [8], so the algorithm proposed for computing secure equilibria does not actually apply for the AG rule. The difference between AG and SE is that AG strategies have to be resiliant to deviations of all the other players, while SE profiles have to be resiliant to deviations by only one player.

In RS, the system is assumed to be monolithic and the environment is made of components that are *partially controllable*. In RS, we search for a profile of strategies where the system ensures its objective and the players that model the environment are given an "*acceptable*" strategy profiles, from which it is assumed that they will not deviate. "Acceptable" is formalized by any *solution concept*, *e.g.* by NE, dominating strategies (Dom), or subgame perfect equilibria (SPE).

Contributions.

1. As a first and central contribution, we propose a novel notion of synthesis where we take into account different players' objectives using the concept of *admissible* strategies [1, 3, 6]. For a player with objective ϕ, a strategy σ is *dominated* by σ' if σ' does as well as σ w.r.t. ϕ against all strategies of the other players, and better for some of those strategies. A strategy σ is *admissible* if it is *not* dominated by another strategy. In [3], the notion of admissibility was lifted to games played on graphs, and algorithmic questions left open were solved in [6], with the goal of *model checking* the set of outcomes that survive the iterative elimination of dominated strategies. Here, we use this notion to derive a meaningful notion to *synthesize* systems with several components using multi-player games, with the following idea. *Rational* players should only play admissible strategies since dominated strategies are clearly *suboptimal*. In *assume-admissible synthesis* (AA), we make the assumption that players play admissible strategies. Then for each player, we search for an admissible strategy that is *winning* against all admissible strategies of other players. AA is *sound*: *any* strategy profile in which each strategy is admissible and winning against admissible strategies of other players, satisfies the objectives of all the players (Theorem 1).

2. We compare different synthesis rules from the literature: First, we apply all the rules on a simple but representative example, and show the main advantages of AA w.r.t. the other rules (Section 4). Then we compare systematically the different approaches to show when a solution for one rule implies a solution for another rule (Figure 5), and we prove that, contrary to other rules, AA yields rectangular sets of solutions (Theorem 8). We argue that the rectangularity property is essential for practical applications.

3. As a third contribution, we provide algorithms to decide the existence of assume-admissible winning strategy profiles and prove the optimal complexity of our algorithm (Theorem 3): PSPACE-complete for Müller, and PTIME for Büchi objectives. We also give an algorithm for the rule AG with multiple players, which was missing in the literature (Theorem 6).

4. As a last important contribution, we provide an abstraction framework which allows us to define sufficient conditions to compute sets of winning assume-admissible strategies for each player in the game compositionally (Theorem 5). The use of state-space abstraction is essential in order to make the methods scale to large systems; we follow the abstract interpretation framework [13, 21]. Moreover, combining abstraction and rectangularity, one can also decompose the problem into smaller problems to be solved for each player. The idea is to look for a strategy profile witnessing the AA rule by computing each strategy separately, which is possible by rectangularity. For each player i, we consider an abstraction of the state space, and give a sufficient condition for finding a strategy for player i by only using computations on the abstract state space. The idea is close to [17] in spirit, but we need a specialized algorithm to approximate the set of admissible strategies. We thus avoid exploring the state space of the original game. This approach is compositional in the following sense: for each player i, a different abstraction can be chosen, which is tailored for player i, and its strategy is computed independently of the other players' strategies. Thus, to find a strategy profile, this abstraction technique is applied to each player one by one, and if all steps succeed in finding strategies, we obtain a strategy profile that satisfies the AA rule.

Additional pointers to related works. We have already mentioned assume-guarantee synthesis [8] and rational synthesis [18, 24]. Those are the closest related works to ours as they pursue the same scientific objective: they propose a framework to synthesize strategy profiles for non-zero sum multi-player games by taking into account the specification of each player. As those works are defined for similar formal setting, we are able to provide formal statements in the core of the paper that add elements of comparison with our work.

In [17], Faella studies several alternatives to the notion of winning strategy including the notion of admissible strategy. His work is for two-players only, and only the objective of one player is taken into account, the objective of the other player is left unspecified. Faella uses the notion of admissibility to define a notion of *best-effort* in synthesis while we use the notion of admissibility to take into account the objectives of the other players in an n player setting where each player has his own objective.

The notion of admissible strategy is definable in strategy logics [11, 27] and decision problems related to the AA rule can be reduced to satisfiability queries in such logics. Nevertheless this would not lead to worst-case optimal algorithms. Based on our previous work [6], we develop in this paper worst-case optimal algorithms.

In [14], Damm and Finkbeiner use the notion of *dominant strategy* to provide a compositional semi-algorithm for the (undecidable) distributed synthesis problem. So while we use the notion of admissible strategy, they use a notion of dominant strategy. The notion of dominant strategy is *strictly stronger*: every dominant strategy is admissible but an admissible strategy is not necessary dominant. Also, in multiplayer games with omega-regular objectives with complete information (as considered here), admissible strategies are always guaranteed to exist [3] while it is not the case for dominant strategies. We will show in an example that the notion of dominant strategy is too strong for our purpose. Also, note that the objective of Damm and Finkbeiner is different from ours: they use dominance as a mean to formalize a notion of *best-effort* for components of a distributed system w.r.t. their common objective, while we use admissibility to take into account the objectives of the other components when looking for a winning strategy for one component to enforce its own objective. Additionally, our formal setting is different from their setting in several respects. First, they consider zero-sum games between a distributed team of players (processes) against a unique environment, each player in the team has the same specification (the specification of the distributed system to synthesize) while the environment is considered as adversarial and so its specification is the negation of the specification of the system. In our case, each player has his *own* objective and we do not distinguish between protagonist and antagonist players. Second, they consider distributed synthesis: each individual process has its own view of the system while we consider games with perfect information in which all players have a complete view of the system state. Finally, let us point out that Damm and Finkbeiner use the term *admissible* for specifications and *not* for strategies (as already said, they indeed consider dominant strategies and not admissible strategies). In our case, we use the notion of *admissible* strategy which is classical in game theory, see e.g. [19, 1]. This vocabulary mismatch is unfortunate but we decided to stick to the term of "admissible strategy" which is well accepted in the literature, and already used in several previous works on (multi-player) games played on graphs [3, 17, 6].

A preliminary version of this work was published in [5].

Structure of the paper. Section 2 contains definitions. In Section 3, we review synthesis rules introduced in the literature and define assume-admissible synthesis. In Section 4, we consider an example; this allows us to underline some weaknesses of the previous rules. Section 5 contains algorithms for Büchi and Müller objectives, and while Section 6 presents the abstraction techniques applied to our rule. Section 7 presents the algorithm for the assume-guarantee rule. Section 8 presents a formal comparison of the different rules.

2 Definitions

2.1 Multiplayer arenas

A *turn-based multiplayer arena* is a tuple $\mathsf{A} = \langle \mathcal{P}, (\mathsf{S}_i)_{i\in\mathcal{P}}, s_{\mathsf{init}}, (\mathsf{Act}_i)_{i\in\mathcal{P}}, \delta\rangle$ where $\mathcal{P}$ is a finite set of players; for $i \in \mathcal{P}$, S_i is a finite set of player i states; we let $\mathsf{S} = \biguplus_{i\in\mathcal{P}} \mathsf{S}_i$; $s_{\mathsf{init}} \in \mathsf{S}$ is the initial state; for every $i \in \mathcal{P}$, Act_i is the set of player i actions; we let $\mathsf{Act} = \bigcup_{i\in\mathcal{P}} \mathsf{Act}_i$; and $\delta\colon \mathsf{S} \times \mathsf{Act} \mapsto \mathsf{S}$ is the transition function. An *outcome* ρ is a sequence of alternating states and actions $\rho = s_1a_1s_2a_2\ldots \in (\mathsf{S}\cdot\mathsf{Act})^\omega$ such that for all $i \geq 1$, $\delta(s_i, a_i) = s_{i+1}$. We write $\rho_i = s_i$, and $\mathsf{act}_i(\rho) = a_i$. A *history* is a finite prefix of an outcome ending in a state. We denote by $\rho_{\leq k}$ the prefix history $s_1a_1\ldots s_k$, and by $\rho_{\geq k}$ the suffix $s_{k+1}a_{k+1}s_{k+2}\ldots$. and write $\mathsf{last}(\rho_{\leq k}) = s_k$, the last state of the history. The set of states *occurring infinitely often* in an outcome ρ is $\mathrm{Inf}(\rho) = \{s \in \mathsf{S} \mid \forall j \in \mathbb{N}.\ \exists i > j, \rho_i = s\}$.

Strategies A *strategy* of player i is a function $\sigma_i : (\mathsf{S}^*\cdot\mathsf{S}_i) \to \mathsf{Act}_i$. A *strategy profile* for the set of players $P \subseteq \mathcal{P}$ is a tuple of strategies, one for each player of P. We write $-i$ for the set $\mathcal{P}\setminus\{i\}$. Let $\Sigma_i(\mathsf{A})$ be the set of the strategies of player i in A, written Σ_i if A is clear from context, and Σ_P the strategy profiles of $P \subseteq \mathcal{P}$. A set $A \subseteq \Sigma_P$ of strategy profiles is *rectangular* if it can be written as $A = \prod_{i\in P} A_i$ where $A_i \subseteq \Sigma_i$.

An outcome ρ is *compatible* with strategy σ for player i if for all $j \geq 1$, $\rho_j \in \mathsf{S}_i$ and $\mathsf{act}_j(\rho) = \sigma(\rho_{\leq j})$. It is compatible with strategy profile $\sigma_\mathcal{P}$ if it is compatible with each σ_i for $i \in \mathcal{P}$. The *outcome of a strategy profile* $\sigma_\mathcal{P}$ is the unique outcome compatible with $\sigma_\mathcal{P}$ starting at s_{init}, denoted $\mathsf{Out}_\mathsf{A}(\sigma_\mathcal{P})$. For any state s, we write $\mathsf{Out}_{\mathsf{A},s}(\sigma_\mathcal{P})$ for the outcome starting at state s. For any history h, we write $\mathsf{Out}_{\mathsf{A},h}(\sigma_{\mathcal{P}})$ for the outcome starting at state $\mathsf{last}(s)$, concatenated to h; formally, $\mathsf{Out}_{\mathsf{A},h}(\sigma_{\mathcal{P}}) = h_{\leq |h|-1} \cdot \mathsf{Out}_{\mathsf{A},\mathsf{last}(h)}(\sigma_{\mathcal{P}})$. Given $\sigma_P \in \Sigma_P$ with $P \subseteq \mathcal{P}$, let $\mathsf{Out}_{\mathsf{A}}(\sigma_P)$ denote the set of outcomes compatible with σ_P, and extend it to $\mathsf{Out}_{\mathsf{A}}(\Sigma')$ where Σ' is a set of strategy profiles. For $E \subseteq \mathsf{S}_i \times \mathsf{Act}_i$, let $\mathsf{Strat}_i(E)$ denote the set of player i strategies σ that only use actions in E in all outcomes compatible with σ.

2.2 Objectives and Games

An *objective* ϕ is a subset of outcomes. An objective is *prefix-independent* if all suffixes of outcomes in ϕ belong to ϕ. Formally, for all outcomes $\rho \in \phi$, for all $k \geq 1$, we have $\rho_{\geq k} \in \phi$. A strategy σ_i of player i is *winning* for objective ϕ_i if for all $\sigma_{-i} \in \Sigma_{-i}$, $\mathsf{Out}_{\mathsf{A}}(\sigma_i, \sigma_{-i}) \in \phi_i$. A *game* is an arena equipped with an objective for each player, written $\mathsf{G} = \langle \mathsf{A}, (\phi_i)_{i \in \mathcal{P}} \rangle$ where for each player i, ϕ_i is an objective. Given a strategy profile σ_P for the set of players P, we write $\mathsf{G}, \sigma_P \models \phi$ if $\mathsf{Out}_{\mathsf{A}}(\sigma_P) \subseteq \phi$. We write $\mathsf{Out}_{\mathsf{G}}(\sigma_P) = \mathsf{Out}_{\mathsf{A}}(\sigma_P)$, and $\mathsf{Out}_{\mathsf{G}} = \mathsf{Out}_{\mathsf{G}}(\Sigma)$. For any coalition $C \subseteq \mathcal{P}$, and objective ϕ, we denote by $\mathsf{Win}_C(\mathsf{A}, \phi)$ the set of states s such that there exists $\sigma_C \in \Sigma_C$ with $\mathsf{Out}_{\mathsf{G},s}(\sigma_C) \subseteq \phi$.

Although we prove some of our results for general objectives, we give algorithms for ω-regular objectives represented by Muller conditions. A Muller condition is given by a family $\mathcal{F}$ of sets of states: $\phi_i = \{\rho \mid \mathrm{Inf}(\rho) \in \mathcal{F}\}$. Following [22], we assume that $\mathcal{F}$ is given by a Boolean circuit whose inputs are S, which evaluates to true exactly on valuations encoding subsets $S \in \mathcal{F}$. We also use linear temporal logic (LTL) [30] to describe objectives. LTL formulas are defined by $\phi := \mathtt{G}\phi \mid \mathtt{F}\phi \mid \mathtt{X}\phi \mid \phi \mathtt{U} \phi \mid \phi \mathtt{W} \phi \mid S$ where $S \subseteq \mathsf{S}$ (we refer to [16] for the semantics). We consider the special case of Büchi objectives, given by $\mathtt{GF}(B) = \{\rho \mid B \cap \mathrm{Inf}(\rho) \neq \varnothing\}$. Boolean combinations of formulas $\mathtt{GF}(S)$ define Muller conditions representable by polynomial-size circuits.

2.3 Dominance

In any game G, a player i strategy σ_i is *dominated* by σ'_i if for all $\sigma_{-i} \in \Sigma_{-i}$, $\mathsf{G}, \sigma_i, \sigma_{-i} \models \phi_i$ implies $\mathsf{G}, \sigma'_i, \sigma_{-i} \models \phi_i$ and there exists $\sigma_{-i} \in \Sigma_{-i}$, such that $\mathsf{G}, \sigma'_i, \sigma_{-i} \models \phi_i$ and $\mathsf{G}, \sigma_i, \sigma_{-i} \not\models \phi_i$, (this is classically called *weak* dominance, but we call it dominance for simplicity). A strategy which is not dominated is *admissible*. Thus, admissible strategies are maximal, and incomparable, with respect to the dominance relation. We write $\mathsf{Adm}_i(\mathsf{G})$ for the set of *admissible* strategies in Σ_i, and $\mathsf{Adm}_P(\mathsf{G}) = \prod_{i \in P} \mathsf{Adm}_i(G)$ the product of the sets of admissible strategies for $P \subseteq \mathcal{P}$.

Strategy σ_i is *dominant* if for all σ'_i, and σ_{-i}, $\mathsf{G}, \sigma'_i, \sigma_{-i} \models \phi_i$ implies $\mathsf{G}, \sigma_i, \sigma_{-i} \models \phi_i$. The set of dominant strategies for player i is written $\mathsf{Dom}_i(\mathsf{G})$. A *Nash equilibrium* for G is a strategy profile $\sigma_{\mathcal{P}}$ such that for all $i \in \mathcal{P}$, and $\sigma'_i \in \Sigma_i$, $\mathsf{G}, \sigma_{-i}, \sigma'_i \models \phi_i$ implies $\mathsf{G}, \sigma_{\mathcal{P}} \models \phi_i$; thus no player can improve its outcome by deviating from the prescribed strategy. A Nash equilibrium for G from s, is a Nash equilibrium for G where the initial state is replaced by s. A *subgame-perfect equilibrium* for G is a strategy profile $\sigma_{\mathcal{P}}$ such that for all histories h, $(\sigma_i \circ h)_{i \in \mathcal{P}}$ is a Nash equilibrium in G from state $\mathsf{last}(h)$, where given a strategy σ, $\sigma \circ h$ denotes the strategy that follows σ starting at history h, i.e. $\sigma \circ h(h') = \sigma(h_{\leq |h|-1} \cdot h')$ if $h'_0 = \mathsf{last}(h)$ and $\sigma \circ h(h') = \sigma(h')$ otherwise.

3 Synthesis Rules

In this section, we review synthesis rules proposed in the literature, and introduce a novel one: the *assume-admissible* synthesis rule (AA). Unless stated otherwise, we fix for this section a game G, with players $\mathcal{P} = \{1, \ldots, n\}$ and their objectives $\phi_1, \ldots, \phi_n$.

Rule Coop**:** The objectives are *achieved cooperatively* if there is a strategy profile $\sigma_{\mathcal{P}} = (\sigma_1, \ldots, \sigma_n)$ such that $\mathsf{G}, \sigma_{\mathcal{P}} \models \bigwedge_{i \in \mathcal{P}} \phi_i$.

This rule [26, 12] asks for a strategy profile that *jointly* satisfies the objectives of all the players. This rule makes *very strong assumptions*: players fully cooperate and strictly follow their respective strategies. This concept is *not robust* against deviations and postulates that the behavior of every component in the system is *controllable*. This weakness is well-known: see e.g. [8] where the rule is called *weak co-synthesis*.

Rule Win**.** The objectives are *achieved adversarially* if there is a strategy profile $\sigma_{\mathcal{P}} = (\sigma_1, \ldots, \sigma_n)$ such that for all $i \in \mathcal{P}$, $\mathsf{G}, \sigma_i \models \phi_i$.

This rule does not require any cooperation among players at all: the rule asks to synthesize for each player i a strategy which enforces his/her objective ϕ_i against all possible strategies of the other players. Strategy profiles obtained by Win are extremely *robust*: each player is able to ensure his/her objective no matter how the other players behave. Unfortunately, this rule is often not applicable in practice: often, none of the players has a winning strategy against *all* possible strategies of the other players. The next rules soften this requirement by taking into account the objectives of other players.

Rule $\mathsf{Win\text{-}under\text{-}Hyp}$: Given a two-player game G with $\mathcal{P} = \{1, 2\}$ in which player 1 has objective ϕ_1, player 2 has objective ϕ_2, player 1 can *achieve adversarially* ϕ_1 under *hypothesis* ϕ_2, if there is a strategy σ_1 for player 1 such that $\mathsf{G}, \sigma_1 \models \phi_2 \rightarrow \phi_1$.

The rule *winning under hypothesis* applies for two-player games only. Here, we consider the synthesis of a strategy for player 1 against player 2 under the hypothesis that player 2 behaves according to his/her specification. This rule is a relaxation of the rule Win as player 1 is *only* expected to win when player 2 plays so that the outcome of the game satisfies ϕ_2. While this rule is often reasonable, it is *fundamentally* plagued by the following problem: instead of trying to satisfy ϕ_1, player 1 could try to falsify ϕ_2, see e.g. [4]. This problem disappears if player 2 has a winning strategy to enforce ϕ_2, and the rule is then safe. We come back to that later in the paper (see Lemma 1).

Assume guarantee Chatterjee et al. in [8] proposed synthesis rules inspired by Win-under-Hyp that avoid the aforementioned problem. The rule was originally proposed in a model with two components and a scheduler. We study here two natural extensions for n players.

Rules $\mathsf{AG}^\wedge$ **and** $\mathsf{AG}^\vee$: The objectives are achieved by

($\mathsf{AG}^\wedge$) *assume-guarantee-*$\wedge$ if there exists a strategy profile $\sigma_{\mathcal{P}}$ such that

1. $\mathsf{G}, \sigma_{\mathcal{P}} \models \bigwedge_{i \in \mathcal{P}} \phi_i$,
2. for all players i, $\mathsf{G}, \sigma_i \models (\bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j) \Rightarrow \phi_i$.

($\mathsf{AG}^\vee$) *assume-guarantee-*$\vee$[1] if there exists a strategy profile $\sigma_{\mathcal{P}}$ such that

1. $\mathsf{G}, \sigma_{\mathcal{P}} \models \bigwedge_{i \in \mathcal{P}} \phi_i$,
2. for all players i, $\mathsf{G}, \sigma_i \models (\bigvee_{j \in \mathcal{P} \setminus \{i\}} \phi_j) \Rightarrow \phi_i$.

[1] This rule was introduced in [7], under the name Doomsday equilibria, as a generalization of the AG rule of [8] to the case of n-players.

The two rules differ in the second requirement: $\mathsf{AG}^\wedge$ requires that player i wins whenever *all* the other players win, while $\mathsf{AG}^\vee$ requires player i to win whenever *one* of the other player wins. Clearly $\mathsf{AG}^\vee$ is stronger, and the two rules are equivalent for two-player games. As shown in [10], for two-player games, a profile of strategy for $\mathsf{AG}^\wedge$ (or $\mathsf{AG}^\vee$) is a Nash equilibrium in a derived game where players want, in lexicographic order, first to satisfy their own objectives, and then as a secondary objective, want to falsify the objectives of the other players. As NE, $\mathsf{AG}^\wedge$ and $\mathsf{AG}^\vee$ require players to *synchronize* on a particular strategy profiles. As we will see, this is not the case for the new rule that we propose.

Rational synthesis [18] and [24] introduce two versions of *rational synthesis* (RS). In the two cases, one of the player, say player 1, models the system while the other players model the environment. The existential version ($\mathsf{RS}^\exists$) searches for a strategy for the system, and a profile of strategies for the environment, such that the objective of the system is satisfied, and the profile for the environment is *stable* according to some solution concept; here we consider the most classical ones, namely, NE, SPE, or Dom. The universal version ($\mathsf{RS}^\forall$) searches for a strategy for the system, such that for all environment strategy profiles that are *stable* according to the solution concept, the objective of the system holds. We write $\Sigma^{\mathsf{NE}}_{G,\sigma_1}$ (resp. $\Sigma^{\mathsf{SPE}}_{\mathsf{G},\sigma_1}$) for the set of strategy profiles $\sigma_{-1} = (\sigma_2, \sigma_3, \ldots, \sigma_n)$ that are NE (resp. SPE) equilibria in the game G when player 1 plays σ_1, and $\Sigma^{\mathsf{Dom}}_{G,\sigma_1}$ for the set of strategy profiles σ_{-1} where each strategy σ_j, $2 \leq j \leq n$, is dominant in the game G when player 1 plays σ_1.

> **Rules** $\mathsf{RS}^{\exists,\forall}(\mathsf{NE},\mathsf{SPE},\mathsf{Dom})$: Let $\gamma \in \{\mathsf{NE},\mathsf{SPE},\mathsf{Dom}\}$, the objective is achieved by:
>
> ($\mathsf{RS}^\exists(\gamma)$) existential rational synthesis under γ if there is a strategy σ_1 of player 1, and a profile $\sigma_{-1} \in \Sigma^\gamma_{\mathsf{G},\sigma_1}$, such that $\mathsf{G}, \sigma_1, \sigma_{-1} \models \phi_1$.
>
> ($\mathsf{RS}^\forall(\gamma)$) universal rational synthesis under γ if there is a strategy σ_1 of player 1, such that $\Sigma^\gamma_{\mathsf{G},\sigma_1} \neq \emptyset$, and for all $\sigma_{-1} \in \Sigma^\gamma_{\mathsf{G},\sigma_1}$, $\mathsf{G}, \sigma_1, \sigma_{-1} \models \phi_1$.

Clearly, ($\mathsf{RS}^\forall(\gamma)$) is stronger than ($\mathsf{RS}^\exists(\gamma)$) and more robust. As $\mathsf{RS}^{\exists,\forall}(\mathsf{NE},\mathsf{SPE})$ are derived from NE and SPE, they require players to synchronize on particular strategy profiles.

Novel rule, assume-admissible We now present our novel rule based on the notion of *admissible strategies.*

> **Rule** AA: The objectives are achieved by *assume-admissible (*AA*) strategies* if there is a strategy profile $\sigma_{\mathcal{P}}$ such that:
>
> 1. for all $i \in \mathcal{P}$, $\sigma_i \in \mathsf{Adm}_i(\mathsf{G})$;
> 2. for all $i \in \mathcal{P}$, $\forall \sigma'_{-i} \in \mathsf{Adm}_{-i}(\mathsf{G})$. $\mathsf{G}, \sigma'_{-i}, \sigma_i \models \phi_i$.

A player-i strategy satisfying conditions 1 and 2 above is called *assume-admissible-winning* (AA*-winning*). A profile of AA-winning strategies is an AA*-winning strategy profile.* The rule AA requires that each player has a strategy *winning* against *admissible* strategies of other players. So we assume that players do not play strategies which are *dominated*, which is reasonable as dominated strategies are clearly *suboptimal options.* Notice that unlike in NE or SPE, players are not required to agree on a given equilibrium profile; they only need to assume the admissibility of the strategies played by other players.

Note that an adversarial environment can be easily considered in the assume-admissible rule: it suffices to add a player with a trivial objective (*i.e.* always winning). The set of admissible strategies will be the whole set of strategies for that player, and other players will then be required to satisfy their objectives against *any* strategy of this player.

The definition of AA does not explicitly require that the strategy profile satisfies all players' objectives; but this is a consequence of the definition:

Proposition 1. *For all* AA*-winning strategy profile* $\sigma_{\mathcal{P}}$*,* $\mathsf{G}, \sigma_{\mathcal{P}} \models \bigwedge_{i \in \mathcal{P}} \phi_i$*.*

Proof. Let $\sigma_{\mathcal{P}}$ be a strategy profile witness of AA. Let i be a player, we have that $\sigma_{-i} \in \mathsf{Adm}_{-i}(\mathsf{G})$, because by Condition 1, for all $j \neq i$, $\sigma_j \in \mathsf{Adm}_j(\mathsf{G})$. Then by Condition 2 we have that $\mathsf{G}, \sigma_{\mathcal{P}} \models \phi_i$. Since this is true for all players i, we have that $\mathsf{G}, \sigma_{\mathcal{P}} \models \bigwedge_{i \in \mathcal{P}} \phi_i$. ☐

The following example shows that AA-winning strategies must be admissible themselves for Proposition 1 to hold.

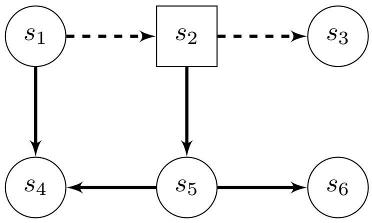

Figure 1: Illustration of the necessity of Condition 1 in the definition of Assume-Admissible Synthesis. Player 1 controls circles and player 2 squares. Player 1 has reachability objective $\phi_1 = \mathtt{F}(s_4 \vee s_6)$ and player 2 reachability objective $\phi_2 = \mathtt{F}(s_4)$.

Example 1. *In* AA*, the profile of strategy must be composed of admissible strategies only. This is necessary as otherwise assumptions of the players on each other may not be satisfied. This is illustrated by the example of Figure 1 in which the two players have reachability objectives* $\phi_1 = \mathtt{F}(s_4 \vee s_6)$ *and* $\phi_2 = \mathtt{F}(s_4)$ *respectively. Admissible strategies are shown in plain edges. Now, the player* 2 *strategy that chooses the dashed edge from* s_2 *satisfies Condition 2 of* AA*, since* s_2 *is not reachable under admissible strategies of player* 1*. Similarly, the player* 1 *strategy that chooses the dashed edge from* s_1 *satisfies Condition 2 of* AA *since the thick edges lead back to a state satisfying* ϕ_1*. But then the resulting profile is such that none of the two players wins.*

4 Synthesis Rules in the Light of an Example

We illustrate the synthesis rules on a multiplayer game which models a real-time scheduler with two tasks. The system is composed of three players, namely, User, Controller, and Scheduler. The high-level description of the system is the following: User sends actions a_1 or a_2 to Controller, which having received action a_i must eventually send a corresponding request r_i to Scheduler. The role of Scheduler is to schedule events: having received r_i, it must issue the event q_i while meeting some temporal constraints.

More precisely, we model the system as a multiplayer game. Accordingly, each round consists of three steps: first, User chooses a valuation for a_1, a_2 (*e.g.* if a_1 is true, then User is sending action a_1), second, Controller chooses a valuation for r_1, r_2, and third, Scheduler chooses a valuation for q_1, q_2. Let us denote by $\perp$ the valuation that assigns false to all variables.

The objective of User is trivial, *i.e.* all outcomes are accepting, since we want the system to accept all sequences of actions made by an arbitrary user. The objectives for Scheduler and Controller are as follows:

1. Upon receiving a_i, Controller must eventually issue r_i within k steps. Moreover, having issued r_i, Controller cannot issue r_i again until the next occurrence of q_i. Doing so, it "filters" the actions issued by User into requests and adheres to constraints imposed by Scheduler.
2. Scheduler is not allowed to schedule the two tasks at the same time. When r_1 is true, then task 1 must be scheduled (q_1) either in the current round or in the next round. When r_2 is true, task 2 must be scheduled (q_2) in the next round.

We will keep k as a parameter.

These requirements can be expressed in LTL as follows:

- $\phi_{\mathsf{User}} = \mathsf{true}$.
- $\phi_{\mathsf{Controller}} = \mathtt{G}(a_1 \Rightarrow \mathtt{F}_{\leq k} r_1) \wedge \mathtt{G}(r_1 \rightarrow \mathtt{X}(\neg r_1 \mathtt{W} q_1)) \wedge \mathtt{G}(a_2 \Rightarrow \mathtt{F}_{\leq k} r_2) \wedge \mathtt{G}(r_2 \rightarrow \mathtt{X}(\neg r_2 \mathtt{W} q_2))$.
- $\phi_{\mathsf{Scheduler}} = \mathtt{G}(r_1 \rightarrow \mathtt{X} q_1 \vee \mathtt{X}^4 q_1) \wedge \mathtt{G}(r_2 \rightarrow \mathtt{X}^4 q_2) \wedge \mathtt{G}\neg(q_1 \wedge q_2)$.

Notice that since each round takes three steps, $X^4 q_2$ (which means $\mathtt{XXXX}q_2$) captures Scheduler's issuing q_i next round. Here, $\mathtt{F}_{\leq k} r_i$ stands for $r_i \vee \mathtt{X} r_i \vee \ldots \vee \mathtt{X}^k r_i$.

Let us call an *action* a_i of User *pending* if Controller has not issued a request r_i since the arrival of a_i. Similarly, we say that a request r_i is pending whenever the corresponding grant q_i has not yet been issued by *Scheduler.*

A solution compatible with the rules proposed in the literature. First, we note that there is no winning strategy neither for Scheduler nor for Controller. In fact, let $\hat{\sigma}_S$ be the strategy of Scheduler that never schedules any of the two tasks, *i.e.* constantly plays $\bot$. Then no Controller strategy is winning against $\hat{\sigma}_S$: if User keeps sending a_i, then Controller can only send r_i once since q_i is never true, thus violating $\phi_{\mathsf{Controller}}$. Second, let $\hat{\sigma}_C$ be a strategy for Controller which always requests the scheduling of both task 1 and task 2, i.e. r_1 and r_2 are constantly true. It is easy to see that this enforces $\neg\phi_{\mathsf{Scheduler}}$ against any strategy of Scheduler. So, there is no solution with rule Win. However strategies $\hat{\sigma}_S$ and $\hat{\sigma}_C$ are clearly not optimal for Scheduler and Controller respectively, since they give up completely on their respective objectives after a deviation while there could be still a chance to satisfy these objectives. Other rules can take into account the objectives to disregard such strategies, so that we may still obtain a solution from the other rules. Observe that the rule $\mathsf{AG}^{\vee}$ has no solution either: in fact, since $\phi_{\mathsf{User}} = \mathsf{true}$, the rule becomes equivalent to Win. Note also that Win-under-Hyp does not apply since we have three players. We now consider a strategy profile which is a solution for the other rules from the literature.

Let (σ_C, σ_S) be strategies for Controller and Scheduler respectively, which behave as follows. At the beginning of each round, given any valuation α on a_1, a_2,

- In the first phase, Controller sends r_1, and Scheduler sends q_1 in the next round, producing a sequence $(\alpha r_1 \bot)(\alpha' \bot q_1)$.
- In the second phase, Controller sends r_2, and Scheduler sends q_2 in the next round, producing $(\alpha r_2 \bot)(\alpha' \bot q_2)$,

Thus, these strategies are independent of User's strategy: whatever the input by User, the same sequence of actions of Controller and Scheduler are prescribed by our strategy. Moreover, if Controller deviates from the above scheme, then Scheduler switches to strategy $\hat{\sigma}_S$ above; and similarly, if Scheduler deviates, Controller switches to $\hat{\sigma}_C$.

This strategy profile is clearly not desirable since it allows for exactly one scenario satisfying the objectives, while under any change in one component's behavior, all objectives fail. Moreover, the outcome does not depend at all on the behavior of User. It is intuitively easy to see that better strategy profiles exist: in fact, both components could continue to "try to satisfy" their objectives in all cases rather than switching to $\hat{\sigma}_C$ or $\hat{\sigma}_S$ which is guaranteed to make all objectives fail. Clearly such pathological strategy profiles should not be solutions to the synthesis problem.

However, we will now show that the rules Coop, $\mathsf{AG}^{\wedge}$, $\mathsf{RS}^{\cdot}(\mathsf{NE}, \mathsf{SPE})$ do allow the above strategy profile:

- Rule Coop: For any σ_U, the outcome of $(\sigma_U, \sigma_C, \sigma_S)$ satisfies all objectives; thus the profile is a possible solution of the rule.
- Rule $\mathsf{AG}^{\wedge}$: When both players follow (σ_C, σ_S), we know that the outcome is a model for both $\phi_{\mathsf{Scheduler}}$ and $\phi_{\mathsf{Controller}}$. We must in addition verify that $\sigma_C \models \phi_{\mathsf{User}} \wedge \phi_{\mathsf{Scheduler}} \rightarrow \phi_{\mathsf{Controller}}$ and that $\sigma_S \models \phi_{\mathsf{User}} \wedge \phi_{\mathsf{Controller}} \rightarrow \phi_{\mathsf{Scheduler}}$. To see the latter property, notice that either the outcome conforms to the above scheme and thus satisfy both objectives, or Controller deviates, in which case Scheduler switches to strategy $\hat{\sigma}_S$ and the outcome satisfies $\neg\phi_{\mathsf{Controller}}$. The argument to show the former property is symmetric.
- Rules $\mathsf{RS}^{\cdot}(\mathsf{NE}, \mathsf{SPE}, \mathsf{Dom})$: We assume that Controller is the system to be synthesized, while User and Scheduler model two components of the environment. We fix σ_C for Controller. In this case, σ_S is a winning strategy for Scheduler. Since ϕ_{User} is trivial, for all σ_U, (σ_U, σ_S) is a solution for NE, SPE, and Dom. Thus the profile is a solution for $\mathsf{RS}^{\exists}(\mathsf{NE}, \mathsf{SPE}, \mathsf{Dom})$. For the universal rules, notice that since σ_S is winning, all dominant strategies for Scheduler are winning too. It follows that all dominant strategies must be identical to $\hat{\sigma}_S$ until a deviation occurs. Thus, under all such strategies Controller's objective is also satisfied. Similarly, Scheduler has a winning strategy in all Nash equilibria and SPE profiles, which satisfy Controller's objective. Thus, the profile is also a solution for $\mathsf{RS}^{\forall}(\mathsf{NE}, \mathsf{SPE}, \mathsf{Dom})$.

Absence of dominant strategies Observe that Controller and Scheduler do not have dominant strategies. Indeed, towards a contradiction, assume that there exists a dominant Controller strategy σ. First, note that the outcome of $(\sigma_U, \sigma, \sigma_S)$ must be identical to the outcome of $(\sigma_U, \sigma_C, \sigma_S)$; in fact, otherwise, this means that σ deviates from σ_C at some point, in which case the outcome is losing for Controller. It follows that $(\sigma_U, \sigma, \sigma_S)$ is losing, while $(\sigma_U, \sigma_C, \sigma_S)$ is winning by definition, so σ cannot be dominant. Consider now strategy σ'_C which is identical to σ_C except that it starts at phase 2 rather than starting at phase 1. One can construct a Scheduler strategy that makes σ'_C win, while making σ_C lose: Scheduler switches to $\hat{\sigma}_S$ in the second round as soon as σ'_C starts being played; and otherwise follows σ_S starting at phase 2. This shows that σ cannot be dominant.

Solutions provided by AA, our novel rule. Let us describe the set of admissible strategies for all players. For Controller we claim that admissible strategies are exactly those strategies σ that satisfy the following conditions for all histories h:

(C0) If $\phi_{\mathsf{Controller}}$ was violated at h, then behave arbitrarily in the rest of the game; otherwise:

(C1) For any $i \in \{1, 2\}$, if r_i is pending at h, then σ sets r_i to false at h.

(C2) For any $i \in \{1, 2\}$, if a_i just became pending at h, then for all histories h' compatible with σ, extending h, and of length $|h| + k$, either r_i is pending at all points $h'_{\leq i}$ with $|h| \leq i \leq |h'|$, or σ sets r_i to true at some history $h'_{\leq i}$ for $|h| \leq i \leq |h'|$.

Any strategy that does not satisfy these conditions is dominated. For instance, if a strategy violates (C1), say at history h, one can obtain a dominating strategy by switching at h to a strategy which respects this safety property. Similarly, if from history h, the strategy never sets r_i in all possible continuations of length k while a_i is pending and r_i is not, one can again modify it by switching to a "better" strategy which does set r_i eventually. The argument is formalized in the following lemma (detailed proofs are given in Appendix A).

Lemma 1. *Any strategy for* Controller *is admissible if, and only if it satisfies (C0), (C1), and (C2) at all histories.*

We now describe the admissible strategies for Scheduler. Consider the set of strategies satisfying the following conditions, at all histories h,

(C3) if both requests r_1 and r_2 were made at the latest round of h, then grant q_1,

(C4) if request r_2 was made in the penultimate round of h, and either r_1 is not pending or the earliest pending request r_1 was made in the latest round, then grant q_2,

(C5) if request r_1 was made in the penultimate round of h and is pending, and r_2 is not pending, or the earliest pending request r_2 was made in the latest round, then grant q_1.

(C6) if both pending requests r_1 and r_2 were made at the penultimate round, then behave arbitrarily in the rest of the game.

Lemma 2. *Any* Scheduler *strategy is admissible if, and only if it satisfies (C3), (C4), (C5), and (C6) at all histories.*

We now show that the rule AA applies in this case: all players' objectives hold under admissible strategies, that is, assuming conditions (C0)–(C6).

Lemma 3. *For all $k \geq 4$, all strategy profiles $(\sigma_U, \sigma_C, \sigma_S)$ satisfying (C1)–(C6) also satisfy $\phi_{\mathsf{User}} \wedge \phi_{\mathsf{Controller}} \wedge \phi_{\mathsf{Scheduler}}$.*

By the way we obtained the solutions of AA, it should be clear that the set of solutions is rectangular. In fact, we independently characterized the set of admissible strategies for Controller, and then for Scheduler, and proved that any combination of these satisfy all objectives.

5 Algorithm for Assume-Admissible Synthesis

In this section, we give an algorithm to decide the assume-admissible rule and to synthesize AA-winning strategy profiles for prefix-independent objectives. Our algorithm is based on the characterization of the outcomes of admissible strategies of [3] and the algorithm of [6] that computes the iterative elimination of dominated strategies. Our general algorithm is an application of these results, but we also improve the complexity analysis in the case of Büchi objectives. The details of the algorithm will be useful in Section 6 where we will adapt the algorithm to abstract state spaces.

5.1 Values and Admissible Outcomes

Let us recall the characterization of the outcomes of admissible strategy profiles given in [6]. We use the game of Figure 2 as a running example for this section. Clearly, none of the players of this game has a winning strategy for his own objective when not taking into account the objective

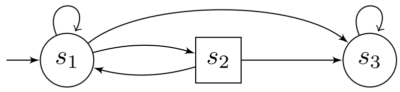

Figure 2: Game G with two players $\mathcal{P} = \{1, 2\}$. Player 1 controls the round states, and has objective $\mathtt{GF}s_2$, and player 2 controls the square state and has objective $\mathtt{GF}s_1$.

of the other player, but, as we will see, both players have an admissible and winning strategy against the *admissible* strategies of the other player, and so the AA rule applies.

The notion of *value* associated to the states of a game plays an important role in the characterization of admissible strategies and their outcomes [3, 6]. We fix a game G. Given a history h, and a set of strategies Σ'_i for player i, we write $\Sigma'_i(h)$ for the set of strategies of Σ'_i compatible with h, that is, the set of strategies σ_i such that h is the prefix of an outcome in $\mathsf{Out}_{\mathsf{G}}(\sigma_i)$. We also write $\Sigma'(h)$ for $\prod_{i \in \mathcal{P}} \Sigma'_i(h)$.

Definition 1 (Value [3])**.** *Let Σ' be a rectangular set of strategy profiles. The* value of history h for player i with respect to Σ', *written* $\mathsf{Val}_i(\Sigma', h)$, *is given by:*

- *if every $\sigma_{\mathcal{P}} \in \Sigma'(h)$ is losing for player i then* $\mathsf{Val}_i(\Sigma', h) = -1$*;*
- *if there is a strategy $\sigma_i \in \Sigma'_i(h)$ such that for all strategy profiles σ_{-i} in $\Sigma'_{-i}(h)$, the profile (σ_i, σ_{-i}) is winning for player i then* $\mathsf{Val}_i(\Sigma', h) = 1$*;*
- *otherwise* $\mathsf{Val}_i(\Sigma', h) = 0$*;*

We use the shorthand notation $\mathsf{Val}_i(h) = \mathsf{Val}_i(\Sigma, h)$. Notice that for prefix-independent objectives, the value only depends on the last state. We may thus write $\mathsf{Val}_i(s) = \mathsf{Val}_i(h)$ for $s = \mathsf{last}(h)$.

A *player j decreases its own value in history h* if there is a position k such that $\mathsf{Val}_j(h_{k+1}) < \mathsf{Val}_j(h_k)$ and $h_k \in \mathsf{S}_j$. We proved in [6], that players do not decrease their own values when playing admissible strategies. In fact, if the current state has value 1, there is a winning strategy which stays within the winning region; if the value is 0, then although other players may force the play into states of value -1, a good strategy for player i will not do this by itself. Let us call those strategies that do not decrease the player's own value *value-preserving.*

Example 2. *In the game of Fig. 2, we have* $\mathsf{Val}_1(s_1) = \mathsf{Val}_1(s_2) = 0$ *and* $\mathsf{Val}_1(s_3) = -1$*; in fact, Player 1 has no winning strategy from any state, and from s_3, it is impossible to satisfy the objective. For Player 2, the situation is similar; we have,* $\mathsf{Val}_2(s_1) = \mathsf{Val}_2(s_2) = 0$ *and* $\mathsf{Val}_2(s_3) = -1$.

Lemma 4 ([6, Lem. 1])**.** *For all games G with prefix-independent objectives, players i, and histories ρ, if* $\mathsf{last}(\rho) \in \mathsf{S}_i$ *and* $\sigma_i \in \mathsf{Adm}_i$ *then* $\mathsf{Val}_i(\delta(\mathsf{last}(\rho), \sigma_i(\rho))) = \mathsf{Val}_i(\rho)$.

We prove here that conversely, any winning outcome on which player i does not decrease its own value is compatible with an admissible strategy of player i. We will use for that three lemmas from [3].

Lemma 5 ([3, Corollary 12], for $\alpha = 1$)**.** *If Σ is non-empty then* Adm *is non-empty.*

Given $\sigma_i, \sigma'_i \in \Sigma_i$, and h such that $\sigma_i(h') = \sigma'_i(h')$ for all prefixes h' of h, $\sigma_i[h \leftarrow \sigma'_i]$ the strategy that agrees with σ'_i on every prefix of h and with σ_i for all other histories. We say that a strategy set Σ_i *allows shifting*, if for any $\sigma_i, \sigma'_i \in \Sigma_i$, such that for all h such that $\sigma_i(h') = \sigma'_i(h')$, $\sigma_i[h \leftarrow \sigma'_i] \in \Sigma_i$. A rectangular set of strategies allows shifting if all its components do.

Lemma 6 ([3, Corollary 10], for $\alpha = 1$)**.** Adm *allows shifting.*

Lemma 7 ([3, Lem. 9])**.** *Let $\Sigma' \subseteq \Sigma$ be a rectangular set that allows shifting. A strategy $\sigma_i \in \Sigma_i$ is admissible if, and only if, the value of $\{\sigma_i\} \times \Sigma_{-i}$ for player i attains or exceeds that of Σ for every reachable history.*

Lemma 8. *Consider game G, a player i, and outcome ρ. If $\rho \models \phi_i$ and player i does not decrease its own value in any prefix of ρ, then there exists a strategy profile $(\sigma_i, \sigma_{-i}) \in \mathsf{Adm}_i \times \Sigma_{-i}$ such that ρ is the outcome of (σ_i, σ_{-i}).*

Proof. We define the strategies σ_i and σ_{-i} such that they precisely follow ρ, but if a deviation from ρ has occurred, they switch to non-dominated strategies. More precisely, if the current history is a prefix $\rho_{\leq k}$ of ρ, then they proceed to the following state ρ_{k+1}. Otherwise there is k such that $h_k = \rho_k$, $h_{k+1} \neq \rho_{k+1}$, and starting from $h_{\leq k+1}$, σ_i follows a non-dominated strategy with respect to $\Sigma(h_{k+1})$. The fact that such non-dominated strategies exists follows from the existence of non-dominated strategies (Lem. 5) and the fact that this set allows shifting (Lem. 6). The outcome ρ is obviously an outcome of this profile. We now have to show that the strategy σ_i is admissible. According to Lem. 7, it is enough to show that for every history h compatible with σ_i, the value for player i with respect to $\{\sigma_i\} \times \Sigma_{-i}$ is greater or equal to its value with respect to Σ.

Let h be a history compatible with σ_i. We distinguish the case where h has deviated from ρ and the case where it has not.

If a deviation has occurred, then σ_i follows a strategy non dominated with respect to $\Sigma(h_{\leq k+1})$ where k is the last index where $h_k = \rho_k$. By Lem. 7, the value of $\{\sigma_i\} \times \Sigma_{-i}(h)$ in h is greater or equal to that of $\Sigma(h_{\leq k+1})$. Since $\Sigma_{-i}(h) \subseteq \Sigma_{-i}(h_{\leq k+1})$, the value of h with respect to $\{\sigma_i\} \times \Sigma_{-i}(h)$ is greater or equal to that with respect to $\Sigma(h)$. Note that by the definition of the value, the value of h with respect to a rectangular set Σ' is equal to that of h with respect to $\Sigma'(h)$. Therefore the value of h with respect to $\{\sigma_i\} \times \Sigma$ is greater or equal to that with respect to Σ.

If a deviation has not occurred then h is a prefix of ρ. The value of h with respect to Σ is greater or equal to 0 since ρ is winning for ϕ_i. Then:

- if the value is 0, then as there is an outcome of σ_i after this history which is winning (the outcome ρ), the value of σ_i is at least 0;
- if the value is 1, then we can show that from history h, σ_i plays a winning strategy: if we stay along ρ, the outcome is winning; if we deviate in a state controlled by player i then since player i does not decrease its own value, the next state has value 1 and σ_i reverts to a winning strategy; otherwise we deviate in a state s of the adversaries, because there is a winning strategy from states of value 1, there is also a winning strategies from all successors of s, so the outcome goes to a state of value 1 and σ_i reverts to a winning strategy.

Therefore the property is satisfied by σ_i and it is admissible.

□

We now introduce some notations to take into account the two previous lemmas in our characterization. We restrict ourselves here to prefix-independent objectives. For player i, let us define the sets $V_{i,x} = \{s \mid \mathsf{Val}_i(s) = x\}$ for $x \in \{-1, 0, 1\}$, which partition S. We define the set of *value-preserving edges* for player i as

$$E_i = \{(s, a) \in \mathsf{S} \times \mathsf{Act} \mid s \in \mathsf{S}_i \Rightarrow \mathsf{Val}_i(\delta(s, a)) = \mathsf{Val}_i(s)\}.$$

Observe that value-preserving strategies for player i are exactly those respecting E_i.

Example 3. *In our running example of Figure 2, it should be clear that any strategy that chooses a transition that goes to s_3 is* not *admissible nor for player 1 neither for player 2. By making such a choice, both players are condemned to lose for their own objectives while other choices would leave a chance to win. In fact, the choice of going to s_3 would decrease their own values. So, we can already conclude that player 2 always chooses $s_2 \mapsto s_1$, which is his only admissible strategy.*

However, not all value-preserving strategies are admissible: *e.g.* for Büchi objectives, staying inside the winning region (that is, states with value 1) does not imply the objective. Moreover, in states of value 0, admissible strategies must visit states where other players can "help" satisfy the objective. Formally, *help states* for player i are other players' states with value 0 and at least two different successors of value 0 or 1. Let us define

$$H_i = \{s \in \mathsf{S} \setminus \mathsf{S}_i \mid \mathsf{Val}_i(s) = 0 \wedge \exists s' \neq s''.\ \{s', s''\} \subseteq \delta(s, \mathsf{Act}) \wedge \mathsf{Val}_i(s') \geq 0 \wedge \mathsf{Val}_i(s'') \geq 0\}.$$

The following lemma, adapted from [6], characterizes the outcomes of admissible strategies. We denote by $\mathtt{G}(E_i)$ the set of outcomes that respect E_i, *i.e.* $\mathtt{G}(\bigvee_{(s,a)\in E_i} s \wedge \mathtt{X}(\delta(s,a)))$.

Lemma 9. *For all games G, and players i, we have $\mathsf{Out}_{\mathsf{G}} \cap \Phi_i = \mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_i, \Sigma_{-i})$, where*

$$\Phi_i = \mathtt{G}(E_i) \wedge (\mathtt{GF}(V_{i,1}) \Rightarrow \phi_i) \wedge (\mathtt{GF}(V_{i,0}) \Rightarrow \phi_i \vee \mathtt{GF}(H_i)).$$

Proof. In [6, Lemma 6], an automaton $\mathcal{A}_i^1$ is defined such that $\mathcal{A}_i^1 \cap \mathsf{Out}_{\mathsf{G}}(\Sigma) = \mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_i, \Sigma_{-i})$. Note that a more general construction $\mathcal{A}_i^n$ was given in [6] but we only need the case $n = 1$ here.

We now analyze further the language of $\mathcal{A}_i^1$. The edges are those of $\mathtt{G}$ except for edges outside of E_i (these edges are noted T in [6]), so the set of outcomes in $\mathcal{A}_i^1$ corresponds to $\mathsf{Out}_{\mathsf{G}} \cap \mathtt{G}(E_i)$. Now an outcome of $\mathcal{A}_i^1$ is accepted if, and only if one the following condition is satisfied, writing $\mathsf{VR}(\rho)$ for the sequence $(\mathsf{Val}(\rho_i))_{i\in\mathbb{N}}$:

- $\mathsf{VR}(\rho) \in 0^*(-1)^\omega$;
- $\mathsf{VR}(\rho) \in 0^*1^\omega$ and $\rho \models \phi_i$;
- $\mathsf{VR}(\rho) \in 0^\omega$ and $\rho \models \phi_i$ or $\rho \models \mathtt{GF}(H_i)$.

Any outcome of $\mathsf{Out}_{\mathsf{G}} \cap \mathtt{G}(E_i)$ reaching some state of value -1 is necessarily losing; thus all successors also have value -1. Similarly, because we removed edges where player i decreases its own value, once the outcomes reaches a state of value 1, it never gets out of these states. Therefore outcomes of $\mathsf{Out}_{\mathsf{G}} \cap \mathtt{G}(E_i)$ have one of the three forms: $0^*(-1)^\omega$, 0^*1^ω or 0^ω.

Let ρ be an outcome that is accepted by $\mathcal{A}_i^1$, it satisfies $\mathtt{G}(E_i)$ and:

- if ρ ends in the states of value -1 then it does not visit $V_{i,1}$ or $V_{i,0}$ infinitely often and thus belongs to Φ_i;
- if ρ ends in the states of value 1, then by the acceptance condition it satisfies ϕ_i and thus belongs to Φ_i;
- otherwise it stays in the states of value 0, then by the acceptance condition, either it satisfies ϕ_i or $\mathtt{GF}(H_i)$ and thus belongs to Φ_i.

Now let ρ be an outcome that satisfies ϕ_i, it satisfies $\mathtt{G}(E_i)$ and therefore corresponds to a valid outcome of $\mathcal{A}_i^1$.

- If ρ ends in the states of value -1 then condition $\mathsf{VR}(\rho) \in 0^*(-1)^\omega$ is satisfied, thus ρ is accepted by $\mathcal{A}_i^1$.

- If ρ ends in the states of value 1, then by definition of Φ_i it satisfies ϕ_i and condition $\mathsf{VR}(\rho) \in 0^*1^\omega$ and $\rho \models \phi_i$ is satisfied, thus ρ is accepted by $\mathcal{A}_i^1$.
- Otherwise it stays in the states of value 0, then by definition of Φ_i, either ϕ_i or $\mathtt{GF}(H_i)$ holds for ρ, hence $\mathsf{VR}(\rho) \in 0^\omega$ and $\rho \models \phi_i$ or $\rho \models \mathtt{GF}(H_i)$ is satisfied, thus ρ is accepted by $\mathcal{A}_i^1$.

This shows that $\Phi_1 \cap \mathsf{Out}_\mathsf{G} = \mathcal{A}_i^1 \cap \mathsf{Out}_\mathsf{G}$ and by [6, Lemma 6], this equals $\mathsf{Out}(\mathsf{Adm}_i, \Sigma_{-i})$. ☐

In our running example of Figure 2, a strategy of player 1 which, after some point, always chooses $s_1 \mapsto s_1$ is dominated by strategies that choose infinitely often $s_1 \mapsto s_2$. This is a corollary of the lemma above. Indeed, while all those strategies only visit states with value 0 (and so do not decrease the value for player 1), the strategy that always chooses $s_1 \mapsto s_1$ has an outcome which is losing for player 1 while the other strategies are compatible with outcomes that are winning for player 1. So, all outcomes of admissible strategies for player 1 that always visit states with values 0, also visits s_2 infinitely often. Using the fact that strategies are value-preserving and the last observation, we can now conclude that both players have (admissible) winning strategies against the admissible strategies of the other players. For instance when player 1 always chooses to play $s_1 \mapsto s_2$, he wins against the admissible strategies of player 2.

5.2 Algorithm for Müller Objectives

For player i, let us define the objective

$$\Omega_i = \mathsf{Out}_\mathsf{G}(\mathsf{Adm}_i) \wedge (\mathsf{Out}_\mathsf{G}(\mathsf{Adm}_{-i}) \Rightarrow \phi_i),$$

which describes the outcomes of admissible strategies of player i, which satisfy objective ϕ_i under the hypothesis that they are compatible with other players' admissible strategies. In fact, it follows from [6] that Ω_i captures the *outcomes* of AA-winning strategies for player i.

Lemma 10. *A player i strategy is* AA*-winning iff it is winning for objective Ω_i.*

Proof. It is shown in [6, Prop. 5] that a strategy of player i is a strategy of Σ_i^n which is winning against all strategies of Σ_{-i}^n if, and only if, it is winning for objective Ω_i^n (where Σ^n is the set of strategies that remain after n steps of elimination, and Ω_i^1 coincides with Ω_i). The results immediately follows from the case $n = 1$. ☐

Thus, solving the AA rule is reduced to solving, for each player i, a game with objective Ω_i. We now give the details of an algorithm with optimal complexity to solve games with these objectives. The algorithm uses procedures from [6], originally developed to compute the *outcomes* that survive the *iterative elimination* of dominated strategies. More precisely, the elimination procedure of [6] first computes the outcomes of admissible strategies; from this it deduces the strategies that are not dominated when all players are restricted to admissible strategies, and their possible outcomes; and this is repeated until the set of outcomes stabilizes. In the end, one obtains the set of the outcomes that are the outcomes of strategy profiles that have survived this iterative elimination. In the rest of this section, we re-visit roughly the first iteration of the above procedure, and explicitly give algorithms to actually synthesize strategies that are winning against admissible strategies.

Objective Ω_i is not prefix-independent since Φ_i has a safety part, thus it cannot be directly expressed as a Müller condition. Since considering prefix-independent objectives simplifies the presentation and the proofs, we are going to encode the information whether $\mathtt{G}(E_i)$, or $\mathtt{G}(\cup_{j\neq i}E_j)$ has been violated in the state space.

Let us decompose Φ_i into $\Phi_i = S_i \wedge M_i$ where $S_i = \mathtt{G}(E_i)$ is a safety condition and

$$M_i = (\mathtt{GF}(V_{i,1}) \Rightarrow \phi_i) \wedge (\mathtt{GF}(V_{i,0}) \Rightarrow (\phi_i \vee \mathtt{GF}(H_i)))$$

is prefix-independent, and can be expressed by a Müller condition described by a circuit of polynomial size.

We now describe the encoding. For each player i, we define game G'_i by taking the product of G with $\{\top, 0, \bot\}$; that is, the states are $\mathsf{S} \times \{\top, 0, \bot\}$, and the initial state $(s_{\mathsf{init}}, 0)$. The transitions are defined as for G for the first component; while from state $(s, 0)$, any action a outside E_i leads to $(\delta(s,a), \bot)$, and any action a outside E_j, for some $j \neq i$, leads to $(\delta(s,a), \top)$. The second component is absorbing at $\bot, \top$. We define

$$\Omega'_i = \left(\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge M'_i \wedge (\wedge_{j\neq i} M'_j \Rightarrow \phi'_i)\right) \vee (\mathtt{GF}(\mathsf{S} \times \{\top\}) \wedge M'_i),$$

where M'_i is the set of outcomes of G'_i whose projections to G are in M_i, and similarly for ϕ'_i.

We will now establish the equivalence of G and G'_i for objectives Ω_i and Ω'_i respectively. Let us first formalize the correspondence between G and G'_i. We define relation $\sim \subseteq \mathsf{S} \times \mathsf{S}'$: for all $(s, x) \in \mathsf{S} \times \{\bot, 0, \top\}$, $s \sim (s, x)$. We extend this to outcomes by $\rho \sim \rho'$ iff for all $i \in \mathbb{N}$, $\rho_i \sim \rho'_i$. The next lemma shows that the relation is a bijection between $\mathsf{Out}_{\mathsf{G}}$ and $\mathsf{Out}_{\mathsf{G}'_i}$.

Lemma 11. *For any $\rho \in \mathsf{Out}_{\mathsf{G}}$ there is a unique $\rho' \in \mathsf{Out}_{\mathsf{G}'_i}$ such that $\rho \sim \rho'$.*

Proof. For any outcome $\rho \in \mathsf{Out}_{\mathsf{G}'_i}$, let us write $\pi(\rho)$ the projection to $\mathsf{Out}_{\mathsf{G}}$ defined by mapping each vertex (s, x) to s.

Assume towards a contradiction that we have ρ' and ρ'' such that $\rho = \pi(\rho') = \pi(\rho'')$. Let k be the last state such that they coincide: $\rho'_k = \rho''_k$ and $\rho'_{k+1} \neq \rho''_{k+1}$. Since $\pi(\rho') = \pi(\rho'')$ we have that they differ only by the second component. We can assume without loss of generality that there are actions a and b such that $(\rho_k, a) \in E_j$ (where player j controls ρ_k), $(\rho_k, b) \notin E_j$ and $\delta(\rho_k, a) = \rho_{k+1} = \delta(\rho_k, b)$. This means that there are actions a and b such that $(s, a) \in E_j$ (where player j controls ρ_k), $(s, b) \notin E_j$ and $\delta(\rho_k, a) = \rho_{k+1} = \delta(\rho_k, b)$. We have $\delta(\rho_k, b) = \delta(\rho_k, a)$, then by definition of E_j and because $(\rho_k, a) \in E_j$, $\mathsf{Val}_j(\delta(s, a)) = \mathsf{Val}_j(s)$ therefore $\mathsf{Val}_j(\delta(s, b)) = \mathsf{Val}_j(s)$ and by definition of E_j, (ρ_k, b) belongs to E_j which contradicts our assumptions and ends the proof. ◻

We thus write π for the bijection which, to $\rho' \in \mathsf{Out}_{\mathsf{G}'_i}$ associates $\rho \in \mathsf{Out}_{\mathsf{G}}$ with $\rho \sim \rho'$. We extend π as a mapping from strategies of G'_i to strategies of G by $\pi(\sigma'_i)(h) = \sigma'_i(\pi^{-1}(h))$. Observe that for all strategies σ'_i, $\pi(\mathsf{Out}_{\mathsf{G}'_i}(\sigma'_i)) = \mathsf{Out}_{\mathsf{G}}(\pi(\sigma'_i))$.

Lemma 12. *Let G be a game, and i a player. Player i has a winning strategy for Ω_i in G if, and only if, he has a winning strategy for Ω'_i in G'_i. Moreover if σ'_i is winning for Ω'_i in G'_i then $\pi(\sigma'_i)$ is winning for Ω_i in G.*

Proof. We will first rewrite Ω_i in a form that is closer to that of Ω'_i. The objective Ω_i is defined by $\mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_i) \cap (\mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_{-i}) \Rightarrow \phi_i)$. Observe that $\mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_{-i}) = \cap_{j\neq i}\mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_j)$ by definition.

$$\Omega_i = \mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_i) \cap \left(\bigcap_{j \neq i} \mathsf{Out}_{\mathsf{G}}(\mathsf{Adm}_j) \Rightarrow \phi_i \right)$$

$$\Omega_i = \Phi_i \cap \mathsf{Out}_{\mathsf{G}} \cap \left(\left(\mathsf{Out}_{\mathsf{G}} \cap \bigcap_{j \neq i} \Phi_j \right) \Rightarrow \phi_i \right) \text{ using Lem. 9}$$

$$\Omega_i = \Phi_i \cap \mathsf{Out}_{\mathsf{G}} \cap \left(\bigcap_{j \neq i} \Phi_j \Rightarrow \phi_i \right)$$

$$\Omega_i = \mathsf{Out}_{\mathsf{G}} \wedge \mathtt{G}(E_i) \wedge M_i \wedge \left(\left(\bigcap_{j \neq i} M_j \Rightarrow \phi_i \right) \vee \bigvee_{j \neq i} \neg \mathtt{G}(E_j) \right)$$

$$\Omega_i = \mathsf{Out}_{\mathsf{G}} \wedge \mathtt{G}(E_i) \wedge M_i \wedge \left(\left(\bigcap_{j \neq i} M_j \Rightarrow \phi_i \right) \vee \bigvee_{j \neq i} \mathtt{F}(\neg E_j) \right)$$

$\boxed{\Rightarrow}$ Let σ_i be a winning strategy for Ω_i in G. We consider the strategy σ'_i defined by $\sigma'_i(h') = \sigma_i(\pi(h'))$ and will show that it is winning for Ω'_i. Let ρ' be an outcome of σ'_i. We have that $\pi(\rho')$ is an outcome of σ_i. Since σ_i is winning for Ω_i, $\pi(\rho')$ belongs to Ω_i.

- If $\pi(\rho') \models M_i \wedge \mathtt{G}(E_i) \wedge \bigvee_{j \neq i} \mathtt{F}(\neg E_j)$, then by construction of δ' the play ρ' reaches a state of $\mathsf{S} \times \{\top\}$ and, from there, only states of $\mathsf{S} \times \{\top\}$ are visited. The condition $\mathtt{GF}(\mathsf{S} \times \{\top\}) \wedge M_i$ is met by ρ' and therefore ρ' is winning for Ω'_i.
- Otherwise $\pi(\rho') \models M_i \wedge \mathtt{G}(E_i) \wedge (\bigwedge_{j \neq i} M_j) \Rightarrow \phi_i$. By construction of δ' the play ρ' stays in $\mathsf{S} \times \{0\}$. The condition $\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge M_i \wedge (\bigwedge_{j \neq i} M_j) \Rightarrow \phi_i$ is met by ρ' and therefore $\rho' \in \Omega'_i$.

This shows that the strategy σ'_i is winning for Ω'_i in G'_i.

$\boxed{\Leftarrow}$ Let σ'_i be a winning strategy for Ω'_i in G'_i, we show that $\pi(\sigma'_i)$ is winning for Ω_i in G. Let ρ be an outcome of $\pi(\sigma'_i)$. We have that $\pi^{-1}(\rho)$ is an outcome of σ'_i. Since σ'_i is winning for Ω'_i, $\pi^{-1}(\rho)$ belongs to Ω'_i. We have that $\pi^{-1}(\rho) \models \mathtt{GF}(\mathsf{S} \times \{0, \top\})$ and by construction of δ' this ensures that all edges that are taken belong to E_i and thus $\pi^{-1}(\rho)$ satisfies the condition $\mathtt{G}(E_i)$.

- If $\pi^{-1}(\rho) \models \mathtt{GF}(S \times \{\top\}) \wedge M_i$ then by construction of δ', an edge outside of E_j for some $j \neq i$ is taken. This ensures condition $\mathtt{F}(\neg E_j)$ and therefore ρ belongs to Ω_i.
- otherwise $\pi^{-1}(\rho) \models (\bigwedge_{j \neq i} M_j \Rightarrow \phi_i)$ and therefore ρ satisfies the condition $\mathtt{G}(E_i) \wedge M_i \wedge \left(\bigcap_{j \neq i} M_j \Rightarrow \phi_i \right)$ and hence belongs to Ω_i.

This shows that the strategy $\pi(\sigma'_i)$ is winning for Ω_i in G.

□

This characterization yields a PSPACE algorithm for checking whether a given player has a AA-winning strategy. In fact, when objectives ϕ_i are given as Müller conditions (described by circuits), the value sets $V_{i,-1}, V_{i,0}, V_{i,1}$ and H_i can be computed in PSPACE. Formulae M_i can be written as circuits of size linear in the size of ϕ_i and the size of the game (in fact, one needs to encode the sets $V_{i,\cdot}$). Condition Ω'_i can also be written in linear size. Last, the game G'_i can be constructed in linear time from G. The algorithm consists in solving G'_i for Player i with objective Ω'_i. Moreover, PSPACE-hardness follows from that of Muller games.

Theorem 1. *Deciding the existence of an* AA*-winning strategy profile is* PSPACE*-complete for Müller objectives.*

Computation of AA-winning Strategy Profiles We just proved the PSPACE-completeness of the *decision* problem; here, we show how to actually *compute* AA-winning strategies. Thanks to Lemma 12, we obtain an algorithm to compute AA-winning strategies by looking for winning strategies in G'_i and projecting them:

Theorem 2. *Given a game* G *with Muller objectives, if* AA *has a solution, then an* AA*-winning strategy profile can be computed in exponential time.*

Proof. If AA has a solution, then by Lemma 12, there is a winning strategy for Ω'_i in G'_i. This Muller game has polynomial size, hence we can compute a winning strategy σ'_i in exponential time (for instance in [29] the authors show that we can compute such a winning strategy via a safety game of size $|\mathsf{S}|!^3$). By Lemma 12, the projection $\pi(\sigma'_i)$ is an AA-winning strategy. Doing this for each player we obtain a strategy profile solution of AA.

□

5.3 Algorithm for Büchi Objectives

In this section, we show that the complexity of the problem can be substantially reduced if players' objective are described by Büchi conditions. In fact, we give a polynomial-time algorithm in this case by showing that Ω'_i is expressible by a parity condition with only four colors.

Theorem 3. *The existence of an* AA*-winning strategy profile can be decided in polynomial time for Büchi objectives.*

The following of this section is devoted to proving this theorem. In the case of Büchi objectives, let us write $\phi_i = \mathtt{GF}(B_i)$ the objective of player i. We can then rewrite the objective M_i defined in Section 5.2 as $M_i = (\mathtt{GF}(V_{i,1}) \Rightarrow \mathtt{GF}(B_i)) \wedge (\mathtt{GF}(V_{i,0}) \Rightarrow (\mathtt{GF}(B_i) \vee \mathtt{GF}(H_i)))$. In game G, an outcome that satisfies $\mathtt{G}(E_i)$ will either visit only $V_{i,1}$ after some point, or only $V_{i,-1}$ after some point, or only $V_{i,0}$ (see the proof of Lemma 9 for details). In order to simplify the notations, and since the propositions $V_{i,1}, V_{i,0}, V_{i,-1}$ are mutually exclusive in the game G, in the following we will only write $V_{i,1}$ to mean $V_{i,1} \wedge \neg V_{i,0} \wedge \neg V_{i,-1}$ (and similarly $V_{i,0}$ and $V_{i,-1}$). We show that M_i coincide with $\mathtt{GF}((V_{i,1} \wedge B_i) \vee (V_{i,0} \wedge B_i) \vee (V_{i,0} \wedge H_i) \vee V_{i,-1})$ on the language $\mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i)$:

$$
\begin{aligned}
\mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap M_i &= \mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap (\mathtt{FG}(V_{i,1}) \cup \mathtt{FG}(V_{i,-1}) \cup \mathtt{G}(V_{i,0})) \cap M_i \\
&= \mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap ((\mathtt{FG}(V_{i,1}) \cap M_i) \cup (\mathtt{G}(V_{i,0}) \cap M_i) \cup (\mathtt{GF}(V_{i,-1}) \cap M_i)) \\
\mathtt{FG}(V_{i,1}) \cap M_i &= \mathtt{FG}(V_{i,1}) \cap \mathtt{GF}(B_i) = \mathtt{FG}(V_{i,1}) \cap \mathtt{GF}(V_{i,1} \cap B_i) \\
\mathtt{FG}(V_{i,0}) \cap M_i &= \mathtt{FG}(V_{i,0}) \cap (\mathtt{GF}(B_i) \cup \mathtt{GF}(H_i)) = \mathtt{FG}(V_{i,0}) \cap \mathtt{GF}(V_{i,0} \cap (B_i \cup H_i)) \\
\mathtt{FG}(V_{i,-1}) \cap M_i &= \mathtt{FG}(V_{i,-1}) \\
\mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap \mathtt{FG}(V_{i,j}) &= \mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap \mathtt{GF}(V_{i,j}) \text{ for all } j \in \{-1, 0, 1\} \\
&\text{Hence:} \\
\mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap M_i &= \mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap \mathtt{GF}(V_{i,1} \cap B_i) \cup \mathtt{GF}((V_{i,0} \cap B_i) \cup (V_{i,0} \cap H_i)) \cup \mathtt{GF}(V_{i,-1}) \\
&= \mathsf{Out}_\mathsf{G}(\Sigma) \cap \mathtt{G}(E_i) \cap \mathtt{GF}((V_{i,1} \cap B_i) \cup (V_{i,0} \cap B_i) \cup (V_{i,0} \cap H_i) \cup V_{i,-1})
\end{aligned}
$$

Therefore M_i coincide with the Büchi condition $\mathtt{GF}(C_i)$ where $C_i = (V_{i,1} \cap B_i) \cup (V_{i,0} \cap B_i) \cup (V_{i,0} \cap H_i) \cup V_{i,-1}$.

We write C'_i for the states of G'_i whose projection is in C_i. We will also write B'_i for the states $B_i \times \{\bot, 0, \top\}$ of the game G'_i.

We define

$$\Omega''_i = (\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge \mathtt{GF}(C'_i) \wedge (\bigwedge_{j \neq i} \mathtt{GF}(C'_j) \Rightarrow \mathtt{GF}(B_i \times \{\bot, 0, \top\}))) \vee (\mathtt{GF}(\mathsf{S} \times \{\top\}) \wedge \mathtt{GF}(C'_i)).$$

Notice that Ω''_i is obtained from Ω'_i by replacing each M'_j by $\mathtt{GF}(C'_j)$. From the observations above, it follows that $\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge \mathtt{GF}(C'_i)$ is equivalent to $\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge M'_i$; however, this is not the case a priori for players $j \neq i$. Nevertheless, we prove in the following lemma that winning for objective Ω''_i in G' is equivalent to winning for the objective Ω_i in G for Player i.

Lemma 13. *Let G be a game, and i a player. Player i has a winning strategy for Ω_i in G if, and only if, he has a winning strategy for Ω''_i in G'_i. Moreover if σ'_i is winning for Ω''_i in G'_i then $\pi(\sigma'_i)$ is winning for Ω_i in G.*

Proof. The proof is very similar to Lem. 12. First as we proved in the proof of Lem. 12, we have that:

$$\Omega_i = \mathsf{Out}_{\mathsf{G}} \wedge \mathtt{G}(E_i) \wedge M_i \wedge \left(\left(\bigcap_{j \neq i} M_j \Rightarrow \phi_i \right) \vee \bigvee_{j \neq i} \mathtt{F}(\neg E_j) \right)$$

We then prove the equivalence.

$\boxed{\Rightarrow}$ Let σ_i be a winning strategy for Ω_i in G. We consider the strategy σ'_i defined by $\sigma'_i(h') = \sigma_i(\pi(h'))$ and will show that it is winning for Ω'_i. Let ρ' be an outcome of σ'_i. We have that $\pi(\rho')$ is an outcome of σ_i. Since σ_i is winning for Ω_i, $\pi(\rho')$ belongs to Ω_i.

- If $\pi(\rho') \models \mathtt{G}(E_i) \wedge \mathtt{GF}(C'_i) \wedge \bigvee_{j \neq i} \mathtt{F}(\neg E_j)$ then it also satisfies $M_i \wedge \mathtt{G}(E_i) \wedge \bigvee_{j \neq i} \mathtt{F}(\neg E_j)$. By construction of δ' the play ρ' reaches a state of $\mathsf{S} \times \{\top\}$ and, from there, only states of $\mathsf{S} \times \{\top\}$ are visited. The condition $\mathtt{GF}(\mathsf{S} \times \{\top\}) \wedge M_i$ is met by ρ'. Therefore ρ' satisfies $\mathtt{GF}(\mathsf{S} \times \{\top\}) \wedge \mathtt{GF}(C'_i)$. It is thus winning for Ω''_i.
- Otherwise $\pi(\rho') \models M_i \wedge \mathtt{G}(E_i) \wedge (\bigwedge_{j \neq i} M_j) \Rightarrow \phi_i$. By construction of δ' the play ρ' stays in $\mathsf{S} \times \{0\}$. The condition $\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge M_i \wedge (\bigwedge_{j \neq i} M_j) \Rightarrow \phi_i$ is met by ρ'. Therefore $\mathtt{GF}(C'_i)$ is also met. Since having $\mathtt{GF}(\mathsf{S} \times \{0\})$ means that no edge outside of E_j is seen for any player j, under the assumption $\mathtt{GF}(\mathsf{S} \times \{0\})$, $\bigwedge_{j \neq i} M_j$ is equivalent to $\bigwedge_{j \neq i} \mathtt{GF}(C'_j)$. Therfore ρ' satisfies $\mathtt{GF}(\mathsf{S} \times \{0\}) \wedge \mathtt{GF}(C'_i) \wedge (\bigwedge_{j \neq i} \mathtt{GF}(C'_j) \Rightarrow \mathtt{GF}(B_i \times \{\bot, 0, \top\}))$. It is thus winning for Ω''_i.

This shows that the strategy σ'_i is winning for Ω''_i in G'_i.

$\boxed{\Leftarrow}$ Let σ'_i be a winning strategy for Ω''_i in G'_i, we show that $\pi(\sigma'_i)$ is winning for Ω_i in G. Let ρ be an outcome of $\pi(\sigma'_i)$. We have that $\pi^{-1}(\rho)$ is an outcome of σ'_i. Since σ'_i is winning for Ω''_i, $\pi^{-1}(\rho)$ belongs to Ω''_i. We have that $\pi^{-1}(\rho) \models \mathtt{GF}(\mathsf{S} \times \{0, \top\})$ and by construction of δ' this ensures that all edges that are taken belong to E_i and thus $\pi^{-1}(\rho)$ satisfies the condition $\mathtt{G}(E_i)$.

- If $\pi^{-1}(\rho) \models \mathtt{GF}(S \times \{\top\}) \wedge \mathtt{GF}(C'_i)$ then by construction of δ', an edge outside of E_j for some $j \neq i$ is taken. This ensures condition $\mathtt{F}(\neg E_j)$ and therefore ρ belongs to Ω_i.
- otherwise $\pi^{-1}(\rho) \models \mathtt{GF}(S \times \{0\}) \wedge \mathtt{GF}(C'_i) \wedge (\bigwedge_{j \neq i} \mathtt{GF}(C'_j) \Rightarrow \mathtt{GF}(B_i \times \{\bot, 0, \top\}))$. Since ρ satisfies condition $\mathtt{G}(E_i)$, it also satisfies M_i (by the main property of C'_i). And since having $\mathtt{GF}(\mathsf{S} \times \{0\})$ means that no edge outside of E_j is seen for any player j, under the assumption $\mathtt{GF}(\mathsf{S} \times \{0\})$, $\bigwedge_{j \neq i} M_j$ is equivalent to $\bigwedge_{j \neq i} \mathtt{GF}(C'_j)$. Therefore ρ satisfies $\mathtt{G}(E_i) \wedge M_i \wedge \left(\bigcap_{j \neq i} M_j \Rightarrow \phi_i \right)$ and hence belongs to Ω_i.

This shows that the strategy $\pi(\sigma'_i)$ is winning for Ω_i in G.

$\square$

Since in game G'_i, states of $\mathsf{S} \times \top$ and $\mathsf{S} \times \bot$ are absorbing (no play can get out of those components) we write an objective equivalent to Ω''_i in terms of runs of G'_i it defines, which is: $(\mathtt{GF}(C'_i \times \{0\}) \wedge (\bigwedge_{j \neq i} \mathtt{GF}(C'_j) \Rightarrow \mathtt{GF}(B_i))) \vee (\mathtt{GF}(C'_i \times \{\top\})$. We define a (small) deterministic parity automaton $\mathcal{A}$ which recognizes this language. Its state space is $(\{s,t,u,v\} \times (\{j \mid j \in \mathcal{P} \setminus \{i\}\} \cup \{f\}))$. Intuitively the first component monitors which of C'_i and B_i occurs infinitely often, and the second component monitors whether we satisfy each of the conditions $\mathtt{GF}(C'_j)$. The transition relation is a product of transitions for the two components: $s \xrightarrow{C'_i \times \{0,\top\}} u$, $u \xrightarrow{\neg C'_i} t$, $\{t,u\} \xrightarrow{C'_i \setminus B'_i} u$, $\{t,u\} \xrightarrow{B'_i} v$, $v \to s$, and $j \xrightarrow{\neg B_j \times \{0\}} j$, $j \xrightarrow{B_j \times \{0\}} j'$ where j' is $j+1$ if $j+1 \in \mathcal{P} \setminus \{i\}$, $j+2$ if $j+1 = i$ and $j+2 \in \mathcal{P}$, f otherwise, $f \to j_0$ where j_0 is the smallest element of $\mathcal{P} \setminus \{i\}$. The structure of the two components of the automaton are represented in Figure 3 and Figure 4. The coloring is defined by a function χ where $\chi(v,*) = 4$ (where $*$ is

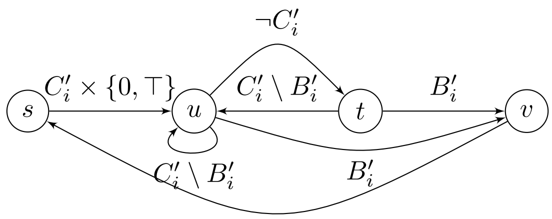

Figure 3: Structure of the first component of automaton $\mathcal{A}$.

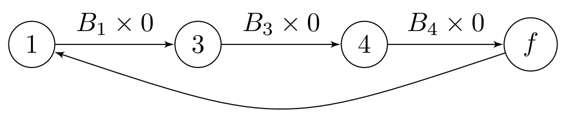

Figure 4: Structure of the second component of automaton $\mathcal{A}$ for 4 players and $i = 2$.

any possible second component), $\chi(\{s,t,u\}, f) = 3$, $\chi(u, \mathcal{P} \setminus \{i\}) = 2$, and for all other states s, $\chi(s) = 1$. A word is accepted by $\mathcal{A}$ when the maximal color appearing infinitely often is even.

We show that a play of G'_i satisfies Ω''_i if, and only if, it is a word accepted by $\mathcal{A}$.

Let ρ be a play of G'_i which satisfies Ω''_i. Either it ends in the $\mathsf{S} \times \top$ component or the $\mathsf{S} \times 0$ component:

- If ρ ends in the $\top$ component then the state of color 3 will not be visited infinitely often, because we need to be in the $\mathsf{S} \times 0$ part of the game to progress on the second component of the automaton. As ρ visits infinitely often C'_i, the corresponding outcome in $\mathcal{A}$ will visit infinitely often u, and therefore the maximal color that appears infinitely often is either 2 or 4.
- Otherwise ρ ends in the 0 component. Since ρ satisfies Ω''_i, it visits C'_i infinitely often and either there is a C'_j for $j \neq i$ that is not visited infinitely often, or ρ visits infinitely often B'_i.
  - If there is a C'_j for $j \neq i$ that is not visited infinitely often, then the second component of $\mathcal{A}$ will get stuck at some point and its state f which is neccessary for color 3, will not be visited infinitely often. As ρ visits infinitely often C'_i, the corresponding outcome in $\mathcal{A}$ will visit infinitely often u, and therefore the maximal color that appears infinitely often is either 2 or 4.

– Otherwise ρ visits infinitely often B'_i. Since we also visit C'_i infinitely often, the outcome of $\mathcal{A}$ corresponding to ρ will reach infinitely often a state $(v, *)$ and therefore the maximal color occurring infinitely often is 4.

This proves that the word is accepted by $\mathcal{A}$.

Now let ρ be a play of G'_i such that the corresponding word is accepted by $\mathcal{A}$. If it is accepted then either the color 4 is seen infinitely often or the color 2 is and the color 3 is not:

- If the color 4 is visited infinitely often then this means t is reached infinitely often in the first component, and because of the structure of $\mathcal{A}$, u also is, which means both $C'_i \times \{0\}$ and B'_i occurs infinitely often. This implies that the outcome ρ belongs to Ω''_i.
- Otherwise the color 2 is visited infinitely often and 3 is not. The states $(*, \top)$ are therefore not visited infinitely often (otherwise the maximal color would be 3 or 4). We deduce from that and the structure of $\mathcal{A}$ that some C'_j for $j \neq i$ is not visited infinitely often. This means $\bigwedge_{j \neq i} \mathtt{GF}(C'_j)$ is not true for ρ. Since the color 2 is seen infinitely often, this means $u, *$ is seen infinitely often and therefore $B_i \times \{0, \top\}$. This ensures ρ belongs to Ω''_i.

This proves that a play of G'_i satisfy Ω''_i if, and only if, it is a word accepted by $\mathcal{A}$.

Then solving the game G'_i with objective Ω''_i is the same as solving it with objective given by $\mathcal{A}$. This can be done by solving the parity game obtained by the product of G'_i with the automaton $\mathcal{A}$. The obtained game is of polynomial size and the number of priorities is 4, such games can be solved in polynomial time (see for instance [25, 31]) and therefore we can decide our problem in polynomial time.

Computation of AA-winning strategies

Theorem 4. *Given a game* G *with Büchi objectives, if* AA *has a solution, then an* AA*-winning strategy profile can be computed in polynomial time.*

Proof. If AA has a solution, then by Lemma 13, there is a winning strategy for Ω''_i in G'_i. This parity game has polynomial size and only 4 priorities. We can compute a winning strategy σ'_i in polynomial time for this kind of games (for instance in [2] the authors compute the most permissive strategy in time $\mathcal{O}(n^{d/2+1})$ where n is the size of the game and d the number of priorities). By Lemma 13, the projection $\pi(\sigma'_i)$ is an AA-winning strategy. Doing this for each player we obtain a strategy profile solution of AA. ☐

Reduction to Strategy Logic As we already mentioned it in the introduction, we can reduce the existence of a winning AA-profile to the model-checking problem of a strategy logic formula [27, 11]. The strategy logic formula is obtained directly from the definition of winning AA-profiles using quantification over strategies and LTL formulas to express the objectives of each player. Remember that the objectives of the players are either succinct Muller objectives defined by circuits, or Büchi objectives defined sets of accepting states, one per player.

To study the complexity of the algorithm that we get from such a reduction, we note that the formula of strategy logic that are construct are of *constant alternation depth* as strategy quantifiers are used exactly as in the definition of winning AA-profiles and so the number of alternation does not depend on the instance of the problem that is considered. On the contrary, the size of the formula which is generated is bounded:

- exponentially in the size of the game graph for succinct Muller games (as – to the best of our knowledge – there does not exist succinct ways to code succinct Muller objectives into LTL objectives),
- bounded polynomially in the size of the game graph times the number of players (as on the contrary Büchi objectives can be coded succinctly in LTL).

Now, if we apply theorem 3 of [11], we get a 2ExpTime algorithm for succinct Muller games and a ExpTime algorithm for Büchi games.

Our results provide better complexities as we provide a PSpace algorithm for succinct Muller games and a PTime algorithm for Büchi games Ñ matching the known lower bounds for the respective problems. Also, we could add that for reachability and safety objectives, easy extension of our solution provides polynomial time algorithms when the number of players is fixed (this is a consequence of Theorem 4 of [6]). Again, those results are out of reach of a direct reduction to strategy logic.

6 Abstraction Framework

We present abstraction techniques to compute assume-admissible strategy profiles following the *abstract interpretation* framework [13]; see [21] for games. Abstraction is a crucial feature for scalability in practice, and we show here that the AA rule is amenable to abstraction techniques. The problem is not directly reducible to computing AA-winning strategies in abstract games obtained as *e.g.* in [15]; in fact, the set of admissible strategies of an abstract game is incomparable with those of the concrete game in general; we give this evidence in Appendix B. Thus, we are going to revisit the assume-admissible synthesis algorithm presented in the previous section, and give an effective sufficient criterion that can be decided solely on the abstract state space.

Overview. Informally, to compute an AA-winning strategy for player k, we construct an abstract game $\mathcal{A}'_k$ with objective $\underline{\Omega}'_k$ s.t. winning strategies of player k in $\mathcal{A}'_k$ map to AA-winning strategies in G. To define $\mathcal{A}'_k$, we re-visit the steps of the algorithm of Section 5 by defining approximations computed on the abstract state space. More precisely, we show how to compute under- and over-approximations of the sets $V_{x,k}$, namely $\underline{V}_{x,k}$ and $\overline{V}_{x,k}$, using fixpoint computations on the abstract state space only. We then use these sets to define approximations of the value preserving edges ($\underline{E}_k$ and $\overline{E}_k$) and those of the help states ($\underline{H}_k$ and $\overline{H}_k$). These are then combined to define objective $\underline{\Omega'}_k$ s.t. if player k wins the abstract game for $\underline{\Omega'}_k$, then he wins the original game for Ω'_k, and thus has an AA-winning strategy.

6.1 Abstract Games.

Consider $\mathsf{G} = \langle \mathsf{A}, (\phi_i)_{i\in\mathcal{P}}\rangle$ with $\mathsf{A} = \langle \mathcal{P}, (\mathsf{S}_i)_{i\in\mathcal{P}}, s_{\mathsf{init}}, (\mathsf{Act}_i)_{i\in\mathcal{P}}, \delta\rangle$ where each ϕ_i is a Müller objective given by a family of sets of states $(\mathcal{F}_i)_{i\in\mathcal{P}}$. Let $\mathsf{S}^\mathsf{a} = \biguplus_{i\in\mathcal{P}} \mathsf{S}^\mathsf{a}_i$ denote a finite set, namely the *abstract state space*. A *concretization function* $\gamma\colon \mathsf{S}^\mathsf{a} \mapsto 2^\mathsf{S}$ is a function such that:

- the abstract states partitions the state space: $\biguplus_{s^\mathsf{a}\in\mathsf{S}^\mathsf{a}} \gamma(s^\mathsf{a}) = \mathsf{S}$,
- it is compatible with players' states: for all players i and $s^\mathsf{a} \in \mathsf{S}^\mathsf{a}_i$, $\gamma(s^\mathsf{a}) \subseteq \mathsf{S}_i$.

We define the corresponding *abstraction* function $\alpha : \mathsf{S} \to \mathsf{S}^\mathsf{a}$ where $\alpha(s)$ is the unique state s^a s.t. $s \in \gamma(s^\mathsf{a})$. We also extend α, γ naturally to sets of states; and to histories, by replacing each element of the sequence by its image.

The pair of abstraction and concretization functions (α, γ) actually defines a *Galois connection*:

Lemma 14. *The pair (α, γ) is a* Galois connection*, that is, for all $S \subseteq \mathsf{S}$ and $T \subseteq \mathsf{S}^{\mathsf{a}}$, we have that $\alpha(S) \subseteq T$ if, and only if, $S \subseteq \gamma(T)$.*

Proof. $\boxed{\Rightarrow}$ Let $s \in S$. Since γ defines a partition of S, there exists $t \in \mathsf{S}^{\mathsf{a}}$ such that $s \in \gamma(t)$. By definition of α, $\alpha(s) = t$. Assuming $\alpha(S) \subseteq T$, we have that $t \in T$. As $s \in \gamma(t)$, we have $s \in \gamma(T)$.

$\boxed{\Leftarrow}$ If $s^{\mathsf{a}} \in \alpha(S)$, then there is $s \in S$ such that $s^{\mathsf{a}} = \alpha(s)$. Assuming $S \subseteq \gamma(T)$, there is $t \in T$ such that $s = \gamma(t)$. By definition of α, we have that $\alpha(s) = t$. Therefore $s^{\mathsf{a}} \in T$.

$\square$

We further assume that γ is *compatible* with all objectives $\mathcal{F}_i$ in the sense that the abstraction of a set S is sufficient to determine whether $S \in \mathcal{F}_i$: for all $i \in \mathcal{P}$, for all $S, S' \subseteq \mathsf{S}$ with $\alpha(S) = \alpha(S')$, we have $S \in \mathcal{F}_i \Leftrightarrow S' \in \mathcal{F}_i$. If the objective ϕ_i is given by a circuit, then the circuit for the corresponding abstract objective ϕ_i^{a} is obtained by replacing each input on state s by $\alpha(s)$. We thus have $\rho \in \phi_i$ if, and only if, $\alpha(\rho) \in \phi_i^{\mathsf{a}}$.

The *abstract transition relation* Δ^{a} induced by γ is defined by:

$$(s^{\mathsf{a}}, a, t^{\mathsf{a}}) \in \Delta^{\mathsf{a}} \Leftrightarrow \exists s \in \gamma(s^{\mathsf{a}}), \exists t \in \gamma(t^{\mathsf{a}}), t = \delta(s, a).$$

We write $\mathsf{post}_\Delta(s^{\mathsf{a}}, a) = \{t^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}} \mid \Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})\}$, and $\mathsf{post}_\Delta(s^{\mathsf{a}}, \mathsf{Act}) = \cup_{a \in \mathsf{Act}} \mathsf{post}_\Delta(s^{\mathsf{a}}, a)$. For each coalition $C \subseteq \mathcal{P}$, we define a game in which players C play together against coalition $-C$; and the former resolves non-determinism in Δ^{a}. Intuitively, the winning region for C in this abstract game will be an over-approximation of the winning region for C in the original game. Given C, the *abstract arena* $\mathcal{A}^C$ is $\langle \{C, -C\}, (\mathsf{S}_C, \mathsf{S}_{-C}), \alpha(s_{\mathsf{init}}), (\mathsf{Act}_C, \mathsf{Act}_{-C}), \delta^{\mathsf{a},C} \rangle$, where

$$\mathsf{S}_C = \left(\bigcup_{i \in C} \mathsf{S}_i^{\mathsf{a}} \right) \cup \left(\bigcup_{i \in \mathcal{P}} \mathsf{S}_i^{\mathsf{a}} \times \mathsf{Act}_i \right), \ \mathsf{S}_{-C} = \bigcup_{i \notin C} \mathsf{S}_i^{\mathsf{a}},$$

and $\mathsf{Act}_C = \left(\bigcup_{i \in C} \mathsf{Act}_i \right) \cup \mathsf{S}^{\mathsf{a}}$ and $\mathsf{Act}_{-C} = \bigcup_{i \in -C} \mathsf{Act}_i$. The relation $\delta^{\mathsf{a},C}$ is given by: if $s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}$, then $\delta^{\mathsf{a},C}(s^{\mathsf{a}}, a) = (s^{\mathsf{a}}, a)$. If $(s^{\mathsf{a}}, a) \in \mathsf{S}^{\mathsf{a}} \times \mathsf{Act}$ and $t^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}$ satisfies $(s^{\mathsf{a}}, a, t^{\mathsf{a}}) \in \Delta^{\mathsf{a}}$ then $\delta^{\mathsf{a},C}((s^{\mathsf{a}}, a), t^{\mathsf{a}}) = t^{\mathsf{a}}$; while for $(s^{\mathsf{a}}, a, t^{\mathsf{a}}) \notin \Delta^{\mathsf{a}}$, the play leads to an arbitrarily chosen state u^{a} with $\Delta(s^{\mathsf{a}}, a, u^{\mathsf{a}})$. Thus, from states (s^{a}, a), coalition C chooses a successor t^{a} which satisfies Δ^{a}.

We extend γ to histories of $\mathcal{A}^C$ by first removing states of $(\mathsf{S}_i^{\mathsf{a}} \times \mathsf{Act}_i)$; and extend α by inserting these intermediate states. Given a strategy σ of player k in $\mathcal{A}^C$, we define its *concretization* as the strategy $\gamma(\sigma)$ of G that, at any history h of G, plays $\gamma(\sigma)(h) = \sigma(\alpha(h))$. We write $\mathsf{Win}_D(\mathcal{A}^C, \phi_k^{\mathsf{a}})$ for the states of S^{a} from which the coalition D has a winning strategy in $\mathcal{A}^C$ for objective ϕ_k^{a}, with $D \in \{C, -C\}$. Informally, it is easier for coalition C to achieve an objective in $\mathcal{A}^C$ than in G, that is, $\mathsf{Win}_C(\mathcal{A}^C, \phi_k^{\mathsf{a}})$ over-approximates $\mathsf{Win}_C(\mathsf{A}, \phi_k)$:

Lemma 15. *If the coalition C has a winning strategy for objective ϕ_k in G from s then it has a winning strategy for ϕ_k^{a} in $\mathcal{A}^C$ from $\alpha(s)$.*

Proof. Assume σ_C is a winning profile of coalition C, for objective ϕ_k in G. We define by induction a winning strategy σ_C^{a} in $\mathsf{G}^{\mathsf{a},k,C}$. We assume that σ_C^{a} has been defined in a manner such that for each finite outcome h^{a} of σ_C^{a} shorter than some bound m, there is some $h \in \gamma(h^{\mathsf{a}})$ such that h is a finite outcome of σ_C. The idea is then to define σ_C^{a} to resolve the determinism in a way which simulates the behavior from h.

- If $s_i^{\mathsf{a}} \in \bigcup_{i \in \mathcal{P}} \mathsf{S}_i^{\mathsf{a}} \times \mathsf{Act}_i$, then $\sigma_C^{\mathsf{a}}(h^{\mathsf{a}} \cdot (\mathsf{last}(h^{\mathsf{a}}), a)) = \gamma(t)$ where $t = \delta(\mathsf{last}(h), a)$.

- If $s^{\mathsf{a}} \in \bigcup_{i\in C} \mathsf{S}^{\mathsf{a}}_i$, $\sigma^{\mathsf{a}}_C(h^{\mathsf{a}} \cdot (\mathsf{last}(h^{\mathsf{a}}), a) \cdot s^{\mathsf{a}}) = \sigma_C(h \cdot \delta(\mathsf{last}(h), a))$.

With this definition, our induction hypothesis will be respected for histories containing one more step, and therefore this holds for all histories. Let now ρ^{a} be an outcome σ^{a}_C. By the way we defined this strategy there is an outcome ρ outcome of σ_C such that $\rho \in \gamma(\rho^{\mathsf{a}})$. As σ_C is winning, ρ satisfies the Muller condition ϕ_k and since γ is compatible with players' objectives, ρ^{a} satisfies ϕ^{a}_k. This shows that C has a winning strategy in $\mathcal{A}^C$ for ϕ^{a}_k.

□

6.2 Value-Preserving Strategies.

We now use the abstract games defined above to define under- and over-approximations for value-preserving strategies for a given player. We start by computing approximations $\underline{V}_{k,x}$ and $\overline{V}_{k,x}$ of the sets $V_{k,x}$, and then use these to obtain approximations of the value-preserving edges E_k (denoted $\underline{E}_k$ and $\overline{E}_k$). At the end of this subsection, we show that these allow us to obtain under- and over-approximations of the set $\gamma(E_k)$ of value-preserving strategies.

Fix a game G, and a player k. Let us define the *controllable predecessors* for player k as

$$\begin{aligned}\mathsf{CPRE}_{\mathcal{A}^{\mathcal{P}\setminus\{k\}},k}(X) = \; & \{s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_k \mid \exists a \in \mathsf{Act}_k, \mathsf{post}_\Delta(s^{\mathsf{a}}, a) \subseteq X\} \\ & \cup\{s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_{\mathcal{P}\setminus\{k\}} \mid \forall a \in \mathsf{Act}_{-k}, \mathsf{post}_\Delta(s^{\mathsf{a}}, a) \subseteq X\}.\end{aligned}$$

We let

$$\begin{aligned}
&\overline{V}_{k,1} = \mathsf{Win}_{\{k\}}(\mathcal{A}^{\{k\}}, \phi^{\mathsf{a}}_k), && \overline{V}_{k,-1} = \mathsf{Win}_\emptyset(\mathcal{A}^\emptyset, \neg\phi^{\mathsf{a}}_k),\\
&\overline{V}_{k,0} = \mathsf{Win}_{\mathcal{P}\setminus\{k\}}(\mathcal{A}^{\mathcal{P}\setminus\{k\}}, \neg\phi^{\mathsf{a},k}) \cap \mathsf{Win}_{\mathcal{P}}(\mathcal{A}^{\mathcal{P}}, \phi^{\mathsf{a}}_k),\\
&\underline{V}_{k,1} = \mathsf{Win}_{\{k\}}(\mathcal{A}^{\mathcal{P}\setminus\{k\}}, \phi^{\mathsf{a}}_k), && \underline{V}_{k,-1} = \mathsf{Win}_\emptyset(\mathcal{A}^{\mathcal{P}}, \neg\phi^{\mathsf{a}}_k)\\
&\underline{V}_{k,0} = \nu X.\big(\mathsf{CPRE}_{\mathcal{A}^{\mathcal{P}\setminus\{k\}},k}(X \cup \underline{V}_{k,1} \cup \underline{V}_{k,-1}) \cap F\big),\\
&\quad\text{where } F = \mathsf{Win}_{\mathcal{P}\setminus\{k\}}(\mathcal{A}^{\{k\}}, \neg\phi^{\mathsf{a}}_k) \cap \mathsf{Win}_{\mathcal{P}}(\mathcal{A}^\emptyset, \phi^{\mathsf{a}}_k).
\end{aligned}$$

The last definition uses the $\nu X.f(X)$ operator which is the greatest fixpoint of f. These sets define approximations of the sets $V_{k,x}$. Informally, this follows from the fact that to define *e.g.* $\overline{V}_{k,1}$, we use the game $\mathcal{A}^{\{k\}}$, where player k resolves itself the non-determinism, and thus has more power than in G. In contrast, for $\underline{V}_{k,1}$, we solve $\mathcal{A}^{\mathcal{P}\setminus\{k\}}$ where the adversary resolves non-determinism. We state these properties formally:

Lemma 16. *For all players k and $x \in \{-1, 0, 1\}$, $\gamma(\underline{V}_{k,x}) \subseteq V_{k,x} \subseteq \gamma(\overline{V}_{k,x})$.*

Proof. $\boxed{\overline{V}_{k,1}}$ This is a direct consequence of Lemma 15.

$\boxed{\overline{V}_{k,-1}}$ If $s \in V_{k,-1}$ then the coalition $\mathcal{P}$ has no winning strategy in G. By determinacy, the empty coalition has a strategy to ensure $\neg\phi_k$. Therefore by Lemma 15, the coalition $\varnothing$ has a strategy in $\mathcal{A}^{\mathcal{P}}$ from $\alpha(s)$ that ensures $\neg\phi_k$. Therefore $s \in \gamma(\overline{V}_{k,-1})$.

$\boxed{\overline{V}_{k,0}}$ Recall that $V_{k,0} = \mathsf{Win}_{\mathcal{P}\setminus\{k\}}(\mathsf{A}, \neg\phi_k) \cap \mathsf{Win}_{\mathcal{P}}(\mathsf{A}, \phi_k)$. Let s be a state in $V_{k,0}$. By Lemma 15, $\alpha(s)$ belongs to both sides of the intersection, thus $\alpha(s) \in \overline{V}_{k,0}$. Thus $V_{k,0} \subseteq \gamma(\overline{V}_{k,0})$.

$\boxed{\underline{V}_{k,1}}$ If $s^{\mathsf{a}} \in \underline{V}_{k,1}$ then the coalition $\mathcal{P} \setminus \{k\}$ has no strategy in $\mathcal{A}^{\mathcal{P}\setminus\{k\}}$ for $\neg\phi^{\mathsf{a}}_k$. Therefore by Lemma 15, it has no strategy in A from any state of $\gamma(s^{\mathsf{a}})$ to do so. Therefore k has a winning strategy in A from $\gamma(s^{\mathsf{a}})$, and $\gamma(s^{\mathsf{a}}) \in V_{k,1}$.

$\boxed{\underline{V}_{k,-1}}$ If $s^{\mathsf{a}} \in \underline{V}_{k,-1}$, then the coalition $\mathcal{P}$ has no winning strategy in $\mathcal{A}^{\mathcal{P}}$ for objective ϕ^{a}_k. Therefore by Lemma 15, it has no winning strategy in A from $\gamma(s^{\mathsf{a}})$ neither for the objective ϕ_k. This means that $\gamma(s^{\mathsf{a}}) \in V_{k,-1}$.

$\boxed{\underline{V}_{k,0}}$ Note that by definition of the $\nu X.$ operator, $\underline{V}_{k,0} \subseteq F$. Thus, let us just show that $\gamma(F) \subseteq V_{k,0}$. Recall that $V_{k,0} = \mathsf{Win}_{\mathcal{P}\setminus\{k\}}(\mathsf{A}, \neg\phi_k) \cap \mathsf{Win}_{\mathcal{P}}(\mathsf{A}, \phi_k)$. Let $s \in \gamma(\underline{V}_{k,0})$. Then player k has no strategy in $\mathcal{A}^{\{k\}}$ for ϕ_k^{a}, hence, by Lemma 15, it cannot win A neither for ϕ_k from $\gamma(s)$. This shows that $\gamma(s) \subseteq \mathsf{Win}_{\mathcal{P}\setminus\{k\}}(\mathsf{A}, \neg\phi_k^{\mathsf{a}})$. Furthermore, the coalition $\emptyset$ has no strategy in $\mathcal{A}^{\mathcal{P}}$ for $\neg\phi_k^{\mathsf{a}}$, thus it does not have one neither in A for $\neg\phi_k$ from $\gamma(s)$. In other terms, $\gamma(s) \subseteq \mathsf{Win}_{\mathcal{P}}(\mathsf{A}, \phi_k)$.

□

We thus have $\cup_x \gamma(\overline{V}_{k,x}) = \mathsf{S}$ (as $\cup_x V_{k,x} = \mathsf{S}$) but this is not the case for $\underline{V}_{k,x}$; so let us define $\underline{V} = \cup_{j\in\{-1,0,1\}} \underline{V}_{k,j}$. We now define approximations of E_k based on the above sets.

$$\begin{aligned}
\overline{E}_k &= \{(s^{\mathsf{a}}, a) \in \mathsf{S}^{\mathsf{a}} \times \mathsf{Act} \mid s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_k \Rightarrow \exists x, s^{\mathsf{a}} \in \overline{V}_{k,x}, \mathsf{post}_\Delta(s^{\mathsf{a}}, a) \cap \cup_{l\geq x} \overline{V}_{k,l} \neq \emptyset\},\\
\underline{E}_k &= \{(s^{\mathsf{a}}, a) \in \mathsf{S}^{\mathsf{a}} \times \mathsf{Act} \mid s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_k \Rightarrow \exists x, s^{\mathsf{a}} \in \underline{V}_{k,x}, \mathsf{post}_\Delta(s^{\mathsf{a}}, a) \subseteq \cup_{l\geq x} \underline{V}_{k,l}\}\\
&\qquad \cup \{(s^{\mathsf{a}}, a) \mid s^{\mathsf{a}} \not\in \underline{V}\}.
\end{aligned}$$

Intuitively, $\overline{E}_k$ is an over-approximation of E_k, and $\underline{E}_k$ under-approximates E_k when restricted to states in $\underline{V}$ (notice that $\underline{E}_k$ contains all actions from states outside $\underline{V}$). In fact, our under-approximation will be valid only inside $\underline{V}$; but we will require the initial state to be in this set, and make sure the play stays within $\underline{V}$. We show that sets $\underline{E}_k$ and $\overline{E}_k$ provide approximations of value-preserving strategies.

We show that when playing according to $\underline{E}_k$, player k ensures staying in $\underline{V}$. This is proven in the following. Let us write $\gamma(\mathcal{E}) = \{(s, a) \mid (\alpha(s), a) \in \mathcal{E}\}$ for $\mathcal{E} \in \{\underline{E}_k, \overline{E}_k\}$.

Lemma 17. *For all games* G*, and players* k*,*

1. $\gamma(\underline{E}_k \cap (\underline{V} \times \mathit{Act})) \subseteq E_k \subseteq \gamma(\overline{E}_k)$.
2. *For all* $s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_k$*, there exist* $a, a' \in \mathit{Act}_k$ *such that* $(s^{\mathsf{a}}, a) \in \underline{E}_k$ *and* $(s^{\mathsf{a}}, a') \in \overline{E}_k$.
3. *For all* $(s^{\mathsf{a}}, a) \in \underline{E}_k$ *with* $s^{\mathsf{a}} \in \underline{V}$*, we have* $\mathit{post}_\Delta(s^{\mathsf{a}}, a) \subseteq \underline{V}$.

Proof. The inclusion $E_k \subseteq \gamma(\overline{E}_k)$ follows from the definition of $\overline{E}_k$, and by Lemma 16. It also follows that for all $s \in \mathsf{S}^{\mathsf{a}}_k$, there is $(s, a') \in \overline{E}_k$, since this is always the case for E_k.

Let (s^{a}, a) be an edge in $\underline{E}_k \cap (\underline{V} \times \mathsf{Act})$. Let s be a state in $\gamma(s^{\mathsf{a}})$. We have that $s \in \gamma(\underline{V}_{k,x})$ for some $x \in \{-1, 0, 1\}$ and by Lemma 16 $s \in V_{k,x}$. By definition of $\underline{E}_k$, for all t^{a} such that $\Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$, $t^{\mathsf{a}} \in \underline{V}_{k,l}$ with $l \geq x$ and $s^{\mathsf{a}} \in \underline{V}_{k,x}$. By Lemma 16, we have that the value of all states in $\gamma(t^{\mathsf{a}})$ are at least as great as any state in $\gamma(s^{\mathsf{a}})$. By definition of Δ, $\alpha(\delta(s, a)) \subseteq \{t^{\mathsf{a}} \mid \Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})\}$. Therefore $\alpha(\delta(s, a)) \in \cup_{l\geq x} \underline{V}_{k,l}$, which means $\delta(s, a) \in \cup_{l\geq x} \gamma(\cup_{l\geq x} \underline{V}_{k,l}) \subseteq \cup_{l\geq x} V_{k,l}$ using Lemma 16. By definition of E_k this implies that $(s, a) \in E_k$.

It remains to prove that for all $s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_k$, there is $(s^{\mathsf{a}}, a) \in \underline{E}_k$, and that if $s^{\mathsf{a}} \in \underline{V}$, then for all $(s^{\mathsf{a}}, a) \in \underline{E}_k$, $\Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ implies $t^{\mathsf{a}} \in \underline{V}$.

If $s^{\mathsf{a}} \in \mathsf{S}^{\mathsf{a}}_k \setminus \underline{V}$, then $(s^{\mathsf{a}}, a) \in \underline{E}_k$ for all $a \in \mathsf{Act}_k$ by definition. Let us now assume $s^{\mathsf{a}} \in \underline{V}$.

- If $s^{\mathsf{a}} \in \underline{V}_{k,-1}$, then By definition of $\underline{V}_{k,-1}$, we have that for all actions a, and all states t^{a}, if $\Delta^{\mathsf{a}}(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ then $t^{\mathsf{a}} \in \underline{V}_{k,-1}$. Thus $(s^{\mathsf{a}}, a) \in \underline{E}_k$, and $t^{\mathsf{a}} \in \underline{V}_{k,-1}$ for any such t^{a}, so $t^{\mathsf{a}} \in \underline{V}$.
- If $s \in \underline{V}_{k,1}$, then there exists a such that $(s^{\mathsf{a}}, a, t^{\mathsf{a}}) \in \Delta^{\mathsf{a}}$ implies $t^{\mathsf{a}} \in \underline{V}_{k,1}$. So $(s^{\mathsf{a}}, a) \in \underline{E}_k$, and $t^{\mathsf{a}} \in \underline{V}_{k,1}$. Moreover this holds for all a with $(s^{\mathsf{a}}, a) \in \underline{E}_k$, since for such a, $(s^{\mathsf{a}}, a, t^{\mathsf{a}}) \in \Delta^{\mathsf{a}}$ implies $t^{\mathsf{a}} \in \underline{V}_{k,1}$ by definition of $\underline{E}_k$.
- If $s \in \underline{V}_{k,0}$, then by the greatest fixpoint defining $\underline{V}_{k,0}$, there exists $a \in \mathsf{Act}_k$ such that for all t^{a} with $\Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$, $t^{\mathsf{a}} \in \underline{V}_{k,0}$. Conversely, for all $(s^{\mathsf{a}}, a) \in \underline{E}_k$, a ensures staying inside $\underline{V}_{k,0} \cup \underline{V}_{k,1}$. Thus for any such a, $(s^{\mathsf{a}}, a) \in \underline{E}_k$, and any t^{a}, $\Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ means $t^{\mathsf{a}} \in \underline{V}_{k,0}$.

$\square$

Recall that $\underline{E}_k$ does not constrain the actions outside the set $\underline{V}$; thus strategies in $\mathsf{Strat}_k(\underline{E}_k)$ can actually choose dominated actions outside $\underline{V}$. To prove that $\mathsf{Strat}_k(\underline{E}_k)$ is an under-approximation of $\mathsf{Strat}_k(E_k)$ when started in $\underline{V}$, we need to formalize the fact that admissible strategies may choose arbitrary actions at states that are not reachable by any outcome. Intuitively, such strategies cannot be dominated since the dominated behavior is never observed.

For any strategy σ, let $\mathsf{Reach}(\mathsf{G}, \sigma)$ denote the set of states reachable from s_{init} by runs compatible with σ. We show that if one arbitrarily modifies an admissible strategy outside the set $\mathsf{Reach}(\mathsf{G}, \sigma)$, the resulting strategy is still admissible.

Lemma 18. *Let σ be a strategy in $\mathsf{Adm}_i(\mathsf{G})$ and σ' a strategy in $\Sigma_i(\mathsf{G})$. If for all histories h such that $\mathsf{last}(h) \in \mathsf{Reach}(\mathsf{G}, \sigma)$, we have $\sigma(h) = \sigma'(h)$, then $\sigma' \in \mathsf{Adm}_i(\mathsf{G})$.*

Proof. For all profiles $\sigma_{-k} \in \Sigma_{-k}(\mathsf{G})$, we have $\mathsf{Out}_{\mathsf{G}}(\sigma_{-k}, \sigma) = \mathsf{Out}_{\mathsf{G}}(\sigma_{-k}, \sigma')$ so if σ' is dominated, then σ would also be dominated, which is a contradiction. $\square$

We now show that the sets $\gamma(\mathsf{Strat}_k(\underline{E}_k))$ and $\gamma(\mathsf{Strat}_k(\overline{E}_k))$ are abstractions of $\mathsf{Strat}_k(E_k)$.

Lemma 19. *For all games G, and players k, $\mathsf{Strat}_k(E_k) \subseteq \gamma(\mathsf{Strat}_k(\overline{E}_k))$, and if $s_{\mathsf{init}} \in \gamma(\underline{V})$, then $\emptyset \neq \gamma(\mathsf{Strat}_k(\underline{E}_k)) \subseteq \mathsf{Strat}_k(E_k)$.*

Proof. Since $E_k \subseteq \gamma(\overline{E}_k)$ by Lemma 17, we have $\mathsf{Strat}_k(E_k) \subseteq \gamma(\mathsf{Strat}_k(\gamma(\overline{E}_k)))$.

Assume $s_{\mathsf{init}} \in \gamma(\underline{V})$. The fact that $\mathsf{Strat}_k(\underline{E}_k)$, thus also $\gamma(\mathsf{Strat}_k(\underline{E}_k))$ are non-empty follows from Lemma 17 too, since for any state s^{a} there is $a \in \mathsf{Act}_k$ with $(s^{\mathsf{a}}, a) \in \underline{E}_k$.

We prove that $\mathsf{Reach}(\mathcal{A}^{\mathcal{P}\setminus\{k\}}, \sigma) \subseteq \underline{V}$ for all $\sigma \in \mathsf{Strat}_k(\underline{E}_k)$. We already know, by Lemma 17, that for all $s^{\mathsf{a}} \in \underline{V}$, if $(s^{\mathsf{a}}, a) \in \underline{E}_k$ then all successors t^{a} with $\Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ satisfies $t^{\mathsf{a}} \in \underline{V}$. We are going to show that for all $s^{\mathsf{a}} \in \underline{V} \cap \mathsf{S}^{\mathsf{a}}_j$ with $j \neq k$, for all $a \in \mathsf{Act}_j$, $\Delta^{\mathsf{a}}(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ implies $t^{\mathsf{a}} \in \underline{V}$.

Consider $s^{\mathsf{a}} \in \underline{V}$. If $s^{\mathsf{a}} \in \underline{V}_{k,1}$, then for all $a \in \mathsf{Act}$, $\Delta^{\mathsf{a}}(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ implies that $t^{\mathsf{a}} \in \underline{V}_{k,1}$, since $\mathcal{P} \setminus \{k\}$ resolves non-determinism. The situation is similar if $s^{\mathsf{a}} \in \underline{V}_{k,-1}$; for all $a \in \mathsf{Act}_j$, $\Delta^{\mathsf{a}}(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ implies $t^{\mathsf{a}} \in \underline{V}_{k,-1}$. If $s^{\mathsf{a}} \in \underline{V}_{k,0}$, then, by the definition of the outer fixpoint, for all $a \in \mathsf{Act}_j$, $\Delta^{\mathsf{a}}(s^{\mathsf{a}}, a, t^{\mathsf{a}})$ implies that $t^{\mathsf{a}} \in \underline{V}$.

Thus $\mathsf{Reach}(\mathcal{A}^{\mathcal{P}\setminus\{k\}}, \sigma) \subseteq \underline{V}$ for all $\sigma \in \mathsf{Strat}_k(\underline{E}_k)$. It then follows that $\mathsf{Reach}(\mathsf{G}, \gamma(\sigma)) \subseteq \gamma(\underline{V})$. So, by Lemma 18, and by the fact that $\gamma(\underline{E}_k) \subseteq E_k$, all strategies in $\gamma(\mathsf{Strat}_k(\underline{E}_k))$ are value preserving, which is to say, belong to $\mathsf{Strat}_k(E_k)$.

$\square$

6.3 Help States

We now define approximations of the help states H_k, where we write $\Delta(s^{\mathsf{a}}, \mathsf{Act}, t^{\mathsf{a}})$ to mean $\exists a \in \mathsf{Act}, \Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$.

$$
\begin{aligned}
\overline{H}_k &= \{s^{\mathsf{a}} \in \overline{V}_{k,0} \setminus \mathsf{S}^{\mathsf{a}}_k \mid \exists t^{\mathsf{a}}, u^{\mathsf{a}} \in \overline{V}_{k,0} \cup \overline{V}_{k,1}.\ \Delta(s^{\mathsf{a}}, \mathsf{Act}, t^{\mathsf{a}}) \wedge \Delta(s^{\mathsf{a}}, \mathsf{Act}, u^{\mathsf{a}})\} \\
\underline{H}_k &= \{s^{\mathsf{a}} \in \underline{V}_{k,0} \setminus \mathsf{S}^{\mathsf{a}}_k \mid \exists a \neq b \in \mathsf{Act}, \mathsf{post}_\Delta(s^{\mathsf{a}}, a) \cap \mathsf{post}_\Delta(s^{\mathsf{a}}, b) = \emptyset, \\
&\qquad \mathsf{post}_\Delta(s^{\mathsf{a}}, a) \cup \mathsf{post}_\Delta(s^{\mathsf{a}}, b) \subseteq \underline{V}_{k,0} \cup \underline{V}_{k,1}\}.
\end{aligned}
$$

Lemma 20. *For all players k, $\gamma(\underline{H}_k) \subseteq H_k \subseteq \gamma(\overline{H}_k)$.*

Proof. Let $s^{\mathsf{a}} \in \underline{H}_k$, and let $a, b \in \mathsf{Act}$ two witnessing actions. For all $s \in \gamma(s^{\mathsf{a}})$, we have $\delta(s, a) \in \gamma(\mathsf{post}_\Delta(s^{\mathsf{a}}, a)) \subseteq V_{k,0} \cup V_{k,1}$ and $\delta(s, b) \in \gamma(\mathsf{post}_\Delta(s^{\mathsf{a}}, a)) \subseteq V_{k,0} \cup V_{k,1}$. Moreover $\alpha(\delta(s, a)) \in \mathsf{post}_\Delta(s^{\mathsf{a}}, a)$, $\alpha(\delta(s, b)) \in \mathsf{post}_\Delta(s^{\mathsf{a}}, b)$, and $\mathsf{post}_\Delta(s^{\mathsf{a}}, a) \cap \mathsf{post}_\Delta(s^{\mathsf{a}}, b) = \varnothing$, therefore $\alpha(\delta(s, a)) \neq \alpha(\delta(s, b))$ and thus $\delta(s, a) \neq \delta(s, b)$. Hence $s \in H_k$.

Now, consider any $s \in H_k$; and let $a, b \in \mathsf{Act}$ be such that $\delta(s,a), \delta(s,b) \in V_{k,0} \cup V_{k,1}$ and $\delta(s,a) \neq \delta(s,b)$. If we write $t^{\mathsf{a}} = \alpha(\delta(s,a))$ and $u^{\mathsf{a}} = \alpha(\delta(s,b))$, then $t^{\mathsf{a}}, u^{\mathsf{a}} \in \overline{V}_{k,0} \cup \overline{V}_{k,1}$, and $\Delta(s^{\mathsf{a}}, a, t^{\mathsf{a}})$, and $\Delta(s^{\mathsf{a}}, b, u^{\mathsf{a}})$; thus $\alpha(s) \in \overline{H}_k$. It follows that $H_k \subseteq \gamma(\overline{H}_k)$. ☐

6.4 Abstract Synthesis of AA-winning strategies.

We now describe the computation of AA-winning strategies in abstract games. Consider game G and assume sets $\underline{E}_i, \overline{E}_i$ are computed for all players i. Roughly, to compute a strategy for player k, we will constrain him to play only edges from $\underline{E}_k$, while other players j will play in $\overline{E}_j$. By Lemma 19, any strategy of player k maps to value-preserving strategies in the original game, and all value-preserving strategies for other players are still present. We now formalize this idea, incorporating the help states in the abstraction.

We fix a player k. We construct an abstract game in which winning for player k implies that player k has an effective AA-winning strategy in G. We define

$$\mathcal{A}'_k = \langle \{\{k\}, -k\}, (\mathsf{S'}^{\mathsf{a}}_k, \mathsf{S'}^{\mathsf{a}}_{-k} \cup \mathsf{S'}^{\mathsf{a}} \times \mathsf{Act}), \alpha(s_{\mathsf{init}}), (\mathsf{Act}_k, \mathsf{Act}_{-k}), \delta_{\mathcal{A}^k} \rangle,$$

where $\mathsf{S'}^{\mathsf{a}} = \mathsf{S}^{\mathsf{a}} \times \{\bot, 0, \top\}$; thus we modify $\mathcal{A}^{\mathcal{P} \setminus \{k\}}$ by taking the product of the state space with $\{\top, 0, \bot\}$. Intuitively, as in Section 5, initially the second component is 0, meaning that no player has violated the value-preserving edges. The component becomes $\bot$ whenever player k plays an action outside of $\underline{E}_k$; and $\top$ if another player j plays outside $\overline{E}_j$ (for $j \in \mathcal{P} \setminus \{i\}$). We extend γ to $\mathcal{A}'_k$ by $\gamma((s^{\mathsf{a}}, x)) = \gamma(s^{\mathsf{a}}) \times \{x\}$, and extend it to histories of $\mathcal{A}'_k$ by first removing the intermediate states $\mathsf{S'}^{\mathsf{a}} \times \mathsf{Act}$. We thus see $\mathcal{A}'_k$ as an abstraction of A' of Section 5.

We define the following approximations of the objectives M'_k and Ω'_k in $\mathcal{A}'_k$.

$$\begin{aligned}
\underline{M'_k} &= (\mathtt{GF}(\overline{V}_{k,1}) \Rightarrow \phi^{\mathsf{a}}_k) \wedge \left(\mathtt{GF}(\overline{V}_{k,0}) \Rightarrow (\phi^{\mathsf{a}}_k \vee \mathtt{GF}(\underline{H}_k))\right), \\
\overline{M}'_k &= (\mathtt{GF}(\underline{V}_{k,1}) \Rightarrow \phi^{\mathsf{a}}_k) \wedge \left(\mathtt{GF}(\underline{V}_{k,0}) \Rightarrow (\phi^{\mathsf{a}}_k \vee \mathtt{GF}(\overline{H}_k))\right), \\
\underline{\Omega'}_k &= \left(\mathtt{GF}(\mathsf{S}^{\mathsf{a}} \times \{0\}) \wedge \underline{M'}_k \wedge \left(\textstyle\bigwedge_{j \neq k} \overline{M'}_j \Rightarrow \phi^{\mathsf{a}}_k\right)\right) \vee (\mathtt{GF}(\mathsf{S}^{\mathsf{a}} \times \{\top\}) \wedge \underline{M'}_k).
\end{aligned}$$

Lemma 21. *We have $\gamma(\underline{M'_k}) \subseteq M'_k \subseteq \gamma(\overline{M'_k})$.*

Proof. We have $\gamma(\phi^{\mathsf{a}}_k) = \phi_k$ by assumption on γ. Thus, by Lemma 16,

$$\gamma((\mathtt{GF}(\overline{V}_{k,1}) \Rightarrow \phi^{\mathsf{a}}_k)) \subseteq \mathtt{GF}(V_{k,1}) \Rightarrow \phi_k) \subseteq \gamma((\mathtt{GF}(\underline{V}_{k,1}) \Rightarrow \phi^{\mathsf{a}}_k)).$$

Similarly, by Lemma 20, we get $\gamma\left(\mathtt{GF}(\overline{V}_{k,0}) \Rightarrow (\phi^{\mathsf{a}}_k \vee \mathtt{GF}(\underline{H}_k))\right) \subseteq \mathtt{GF}(V_{k,0} \Rightarrow (\phi_k \vee \mathtt{GF}(H_k)) \subseteq \gamma\left(\mathtt{GF}(\underline{V}_{k,0}) \Rightarrow (\phi^{\mathsf{a}}_k \vee \mathtt{GF}(\overline{H}_k))\right)$. It follows that $\gamma(\underline{M'}_k) \subseteq M'_k \subseteq \gamma(\overline{M'}_k)$.
☐

The following lemma implies our main result, stated next as a theorem.

Lemma 22. *Let $k \in \mathcal{P}$ be a player and σ_k a strategy of player k. If $s^{\mathsf{a}}_{\mathsf{init}} \in \underline{V}$, and σ_k is winning for objective $\underline{\Omega'}_k$ in $\mathcal{A}'_k$, then $\gamma(\sigma_k)$ is winning for Ω'_k in G'_k.*

Proof. Let us rewrite

$$\underline{\Omega'}_i = \underline{M'}_i \wedge \left(\left(\mathtt{GF}(\mathsf{S}^{\mathsf{a}} \times \{0\}) \wedge (\bigwedge_{j \neq i} \overline{M'}_j \Rightarrow \phi^{\mathsf{a}}_i)\right) \vee \mathtt{GF}(\mathsf{S}^{\mathsf{a}} \times \{\top\})\right).$$

Let σ_k be a winning strategy in $\mathcal{A}'_k$ for $\underline{\Omega'}_k$. We will show that $\mathsf{G}'_k, \gamma(\sigma_k) \models \Omega'_k$.

Consider any run ρ of G'_k compatible with $\gamma(\sigma_k)$. By definition of $\gamma(\sigma_k)$, $\alpha(\rho)$ is an outcome of $\mathcal{A}'_k$ compatible with σ_k. Since σ_k is a winning strategy, $\alpha(\rho) \in \underline{M'}_k$, and by Lemma 21 $\rho \in M'_k$.

We now show that $\rho \in \mathtt{GF}(\mathsf{S} \times \{0, \top\})$. By assumption, we have $\mathcal{A}'_k, \sigma_k \models \mathtt{GF}(\mathsf{S}^{\mathsf{a}} \times \{0, \top\})$, which means that for all histories h^{a} of $\mathcal{A}'_k$ compatible with σ_k, $(\mathsf{last}(h^{\mathsf{a}}), \sigma(h^{\mathsf{a}})) \in \underline{E}_k$ (otherwise the transition relation of $\mathcal{A}'_k$ would lead to a $\perp$ state). Moreover, since $s^{\mathsf{a}}_{\mathsf{init}} \in \underline{V}$, it follows from Lemma 19 that $(\mathsf{last}(h), \gamma(\sigma)(h)) \in E_k$ for all histories h compatible with $\gamma(\sigma_k)$. Thus no state $(*, \perp)$ is reachable under $\gamma(\sigma)$ in G'_k.

Because of the structure of G'_k this means that ρ either visits states of $\mathsf{S} \times \{0\}$ or states of $\mathsf{S} \times \{\top\}$ infinitely often:

- If $\rho \in \mathtt{GF}(\mathsf{S} \times \{0\})$, then $\alpha(\rho) \in \mathtt{GF}(\mathsf{S}^{\mathsf{a}} \times \{0\})$; so $\alpha(\rho) \in \bigwedge_{j \neq k} \overline{M'}_j \Rightarrow \phi^{\mathsf{a}}_k$; it follows, by Lemma 21 and the compatibility of the abstraction with players' objectives, that $\rho \in \bigwedge_{j \neq k} M'_j \Rightarrow \phi_k$. Thus $\rho \in \Omega'_k$.
- Otherwise $\rho \in \mathtt{GF}(\mathsf{S} \times \{\top\})$, so $\rho \in \Omega'_k$.

Thus any outcome ρ of $\gamma(\sigma_k)$ belongs to Ω'_k which shows it is winning. □

Theorem 5. *For all games G, and players k, if $s_{\mathsf{init}} \in \underline{V}$, and player k has a winning strategy in $\mathcal{A}'_k$ for objective $\underline{\Omega}'_k$, then he has a winning strategy in G'_k for Ω_k; and thus a AA-winning strategy in G.*

Theorem 5 allows one to find AA-winning strategies using abstractions. In fact, for each player k, one can define an abstraction, construct and solve the game $\mathcal{A}'_k$ for objective $\underline{\Omega}'_k$. If this succeeds for each player k, the obtained strategies yield an AA-winning strategy profile in G.

7 Algorithm for Assume-Guarantee Synthesis

The assume-guarantee-$\wedge$ rule was studied in [8] for particular games with three players. However, the given proofs are based on *secure equilibria* which do not actually capture assume-guarantee synthesis, so the correctness of the algorithm of [8] is not clear. We first give an example that illustrates the non-correspondance of secure equilibria and assume-guarantee synthes, and then give an alternative algorithm for deciding assume-guarantee-$\wedge$ for multiplayer games, and prove its correctness.

We recall that a *secure equilibrium* [8] is a strategy profile $\sigma_{\mathcal{P}}$ such that for any player i, and $\sigma'_i \in \Sigma_i$, $\mathsf{Out}(\sigma'_i, \sigma_{-i}) <_i \mathsf{Out}(\sigma_{\mathcal{P}})$ where $\rho <_i \rho'$ means $\rho \not\models \phi_i \wedge \rho' \models \phi_i$ or $\rho' \models \phi_i \wedge |\{j \neq i \mid \rho \models \phi_j\}| < |\{j \neq i \mid \rho' \models \phi_j\}|$.

Example 4. *We consider a game with three players: player 1 controls the valuation of x_1; player 2 the valuation of x_2, and player 3 is a scheduler which gives turn to either player 1 or player 2 at each step. player 3 is assumed to be fair in the sense that at every point in the game each player eventually gets to play. Consider the following objective for player 1: $\phi_1 = (x_2 \to \mathtt{X} x_1) \wedge (\mathtt{F} x_1 \to \mathtt{F} x_2)$. The objective for player 2 is trivial (always true). We consider strategy σ_3 of player 3 that alternates between each player. Strategy σ_1 of player 1 puts x_1 to true once x_2 has been put to true at least once. Strategy σ_2 of player 2 never puts x_2 to true. These strategies form a secure equilibrium which satisfies each objective since we cannot improve the outcome with respect to $<_i$ by changing only the strategy of player i. However it is not an assume-guarantee solution: if we consider another scheduler strategy σ'_3 which gives twice the turn to player 2, and a strategy σ'_2 which will put x_2 to true in the first turn, then $\mathsf{Out}(\sigma_1, \sigma'_2, \sigma'_3) \not\models \phi_1$. The same is in fact true for any strategy σ'_1 of player 1 so there is no assume-guarantee synthesis solution, which contradicts [8, Thm. 4].*

We now give an algorithm for assume-guarantee synthesis. For any game G, and state s, we denote by G_s the game obtained by making s the initial state. Assuming that each player i has an objective ϕ_i which is prefix independent, let us define $W_i = \{s \in \mathsf{S} \mid \exists \sigma_i.\ G_s, \sigma_i \models \bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j \Rightarrow \phi_i\}$.

The following lemma gives a decidable characterization of assume-guarantee synthesis:

Lemma 23. *Let $(\phi_i)_{i \in \mathcal{P}}$ be prefix-independent objectives. Rule $\mathsf{AG}^\wedge$ has a solution if, and only if, there is an outcome ρ which visits only states of $\bigcap_{i \in \mathcal{P}} W_i$ and such that $\rho \models \bigwedge_{i \in \mathcal{P}} \phi_i$.*

Proof. $\boxed{\Rightarrow}$ Let $\sigma_{\mathcal{P}}$ be a solution of $\mathsf{AG}^\wedge$. Let ρ be its outcome. We have that $\rho \models \bigwedge_{i \in \mathcal{P}} \phi_i$ by hypothesis of $\mathsf{AG}^\wedge$. Let i be a player, we show that ρ only visits states of W_i. This is because σ_i is winning for $\bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j \Rightarrow \phi_i$. For all k, $\rho_{\leq k}$ is a finite outcome of σ_i, and the strategy played by σ_i after this history is winning for $\bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j \Rightarrow \phi_i$, which means that ρ_k belongs to W_i. Hence ρ satisfies the desired conditions.

$\boxed{\Leftarrow}$ If there is such an outcome ρ, we define the strategy profile $\sigma_{\mathcal{P}}$ to follow this outcome if no deviation has occurred and otherwise each player i plays a strategy which is winning for $\bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j \Rightarrow \phi_i$ if possible. We show that such a strategy profile satisfies the assumption of assume-guarantee. Obviously $\sigma_{\mathcal{P}} \models \bigwedge_{i \in \mathcal{P}} \phi_i$. Let ρ' be an outcome of σ_i and k the first index such that $\rho'_k \neq \rho_k$. The state $\rho'_{k-1} = \rho_{k-1}$ is not controlled by player i, because σ_i follows ρ. As ρ_{k-1} is in W_i and not controlled by player i, this means that $\rho'_k \in W_i$. Therefore σ_i plays a winning strategy from ρ'_k for the objective $\bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j \Rightarrow \phi_i$; thus ρ' satisfies this objective. Hence $\sigma_{\mathcal{P}}$ is a solution of $\mathsf{AG}^\wedge$. ◻

We deduce a polynomial-space algorithm for the $\mathsf{AG}^\wedge$ rule with Muller objectives:

Theorem 6. *For multiplayer games with Muller objectives, deciding whether $\mathsf{AG}^\wedge$ has a solution is PSPACE-complete.*

Proof. The algorithms proceed by computing the set W_i for each player i with an algorithm that computes winning regions and then checks whether there is an outcome in the intersection $\bigcap_{i \in \mathcal{P}} W_i$ which satisfies $\bigwedge_{i \in \mathcal{P}} \phi_i$. This algorithm is correct thanks to Lemma 23.

This is in PSPACE because the objective $\bigwedge_{j \in \mathcal{P} \setminus \{i\}} \phi_j \Rightarrow \phi_i$ can be expressed by a Muller condition encoded by a circuit [23] of polynomial size. We can decide in polynomial space if a given state is winning for a Muller condition given by a circuit. Thus, the set $\bigcap_{i \in \mathcal{P}} W_i$ can be computed in polynomial space; let us denote by G' the game restricted to this set. The algorithm then consists in finding an outcome in G' satisfying $\bigwedge_{i \in \mathcal{P}} \phi_i$; that is, finding an outcome satisfying a Muller condition, which can be done in polynomial space. ◻

8 Comparison of Synthesis Rules

In this section, we compare the synthesis rules to understand which ones yield solutions more often, and to assess their robustness. Some relations are easy to establish; for instance, rules $\mathsf{Win}, \mathsf{AG}^\vee, \mathsf{AG}^\wedge, \mathsf{AA}$ imply Coop by definition (and Theorem 1). We summarize the implication relations between the rules in Figure 5. A plain arrow from A to B means that if A has a solution, then so does B; while a dashed arrow with a cross means that this implication does not hold. We use some shortcuts for groups of rules: the arrow from Win to the group $\mathsf{RS}^\forall(\cdot)$ means that Win implies all of them. The dashed arrow from the whole group of $\mathsf{RS}^{\forall,\exists}(\cdot)$ to Coop means that none of the rules in the box implies Coop. References to lemmas that prove the relations are given on each arrow. Missing arrows are either trivial relations or they are open; note that some relations can be deduced by transitivity (e.g. Win implies $\mathsf{AG}^{\wedge}$). Note that an arrow does not imply an inclusion between the witnessing strategy profiles.

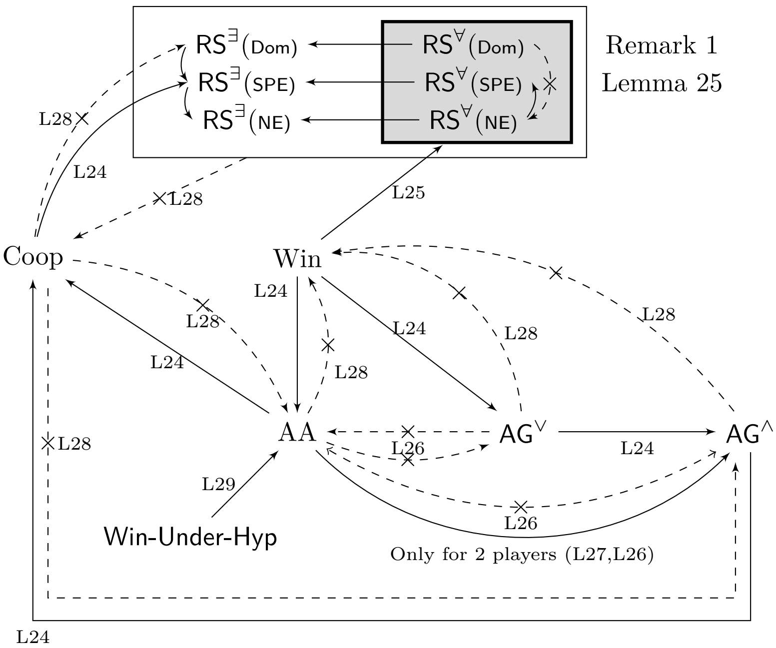

Figure 5: Comparison of synthesis rules.

The following theorem states the correctness of our diagram.

Theorem 7. *The implication relations of Figure 5 hold.*

We will present the proof of each comparison of the diagram in Fig. 5.

Remark 1. *We have* $\mathsf{RS}^{\exists}(\mathsf{SPE}) \Rightarrow \mathsf{RS}^{\exists}(\mathsf{NE})$ *and* $\mathsf{RS}^{\forall}(\mathsf{NE}) \Rightarrow \mathsf{RS}^{\forall}(\mathsf{SPE})$ *because any subgame perfect equilibrium is also a Nash equilibrium. Moreover, in the definition of the rules* RS*, the conditions for* $\mathsf{RS}^{\forall}$ *are stronger than for* $\mathsf{RS}^{\exists}$*, so* $\mathsf{RS}^{\forall}(\mathsf{SPE}) \Rightarrow \mathsf{RS}^{\exists}(\mathsf{SPE})$*,* $\mathsf{RS}^{\forall}(\mathsf{NE}) \Rightarrow \mathsf{RS}^{\exists}(\mathsf{NE})$ *and* $\mathsf{RS}^{\forall}(\mathsf{Dom}) \Rightarrow \mathsf{RS}^{\exists}(\mathsf{Dom})$*.*

Lemma 24. $\mathsf{Win} \Rightarrow \mathsf{AA} \Rightarrow \mathsf{Coop} \Rightarrow \mathsf{RS}^{\exists}(\mathsf{SPE})$ *and* $\mathsf{Win} \Rightarrow \mathsf{AG}^{\vee} \Rightarrow \mathsf{AG}^{\wedge} \Rightarrow \mathsf{Coop}$*,*

Proof. $\boxed{\mathsf{Win} \Rightarrow \mathsf{AA}}$ This holds because winning strategies are always admissible [3], therefore a profile witness of Win satisfies condition 1 and 2 of the definition of assume-admissible.

$\boxed{\mathsf{AA} \Rightarrow \mathsf{Coop}}$ This holds by Theorem 1.

$\boxed{\mathsf{Coop} \Rightarrow \mathsf{RS}^{\exists}(\mathsf{SPE})}$ Note that in order for RS to make sense we must have $\mathsf{sys} \in \mathcal{P}$. Assume Coop has a solution and let $\sigma_{\mathcal{P}}$ be a profile of strategy such that for all player i, $\sigma_{\mathcal{P}} \models \phi_i$.

We define a strategy profile σ'_i, that follows the path $\rho = \mathsf{Out}_{\mathsf{G}}(\sigma_i)$ when possible (that is: if h is a prefix of ρ then play $\mathsf{act}_{|h|}(\rho)$) and if not follows a subgame perfect equilibrium: that is, we select for each state s a subgame perfect equilibrium $\sigma^s_{\mathcal{P}}$, there always exist one for Borel games (so in particular for Muller games) [32, Theorem 3.15]; then if h is not a prefix of ρ, let j be the last index such that $h_{\le j} = \rho_{\le j}$ and we define $\sigma'_{\mathcal{P}}(h) = \sigma^{h_{j+1}}_{\mathcal{P}}(h_{\ge j+1})$.

Let h be a history. If h is a prefix of ρ then the objective of each player is satisfied by following $\sigma'_i \circ h$ so none of them can gain by changing its strategy, therefore it is a Nash equilibrium from $\mathsf{last}(h)$. If h is not a prefix of ρ then by definition of σ'_i, players follow a subgame-perfect equilibrium since h deviated from ρ, so in particular $\sigma'_i \circ h$ is a Nash equilibrium from $\mathsf{last}(h)$. Moreover the objective of the system is satisfied. Therefore $\sigma_{\mathcal{P}}$ is a solution to $\mathsf{RS}(\mathsf{SPE})$.

$\boxed{\mathsf{Win} \Rightarrow \mathsf{AG}^\vee}$ Let $\sigma_{\mathcal{P}}$ such that for each player i, σ_i is winning for ϕ_i. The first condition in the definition of $\mathsf{AG}^\vee$ is satisfied because for all player i, $\mathsf{Out}_\mathsf{G}(\sigma_{\mathcal{P}})$ satisfies ϕ_i. The second condition is satisfied because for all strategy σ'_{-i}, we have that $\mathsf{Out}_\mathsf{G}(\sigma_i, \sigma'_{-i})$ satisfies ϕ_i, so in particular it satisfies $(\bigvee_{j\in\mathcal{P}\setminus\{i\}} \phi_j \Rightarrow \phi_i)$. Hence $\sigma_{\mathcal{P}}$ is a solution for $\mathsf{AG}^\vee$.

$\boxed{\mathsf{AG}^\vee \Rightarrow \mathsf{AG}^\wedge}$ This holds because the second condition in the definition of these rules is stronger for $\mathsf{AG}^\vee$.

$\boxed{\mathsf{AG}^\wedge \Rightarrow \mathsf{Coop}}$ This implication holds simply because of the condition 1 in the definition of assume-guarantee, which corresponds to the definition of Cooperative synthesis.

□

Lemma 25. *For all* $\gamma \in \{\mathsf{NE}, \mathsf{SPE}, \mathsf{Dom}\}$*,* $\mathsf{Win} \Rightarrow \mathsf{RS}^\forall(\gamma)$*,* $\mathsf{RS}^\forall(\mathsf{Dom}) \not\Rightarrow \mathsf{RS}^\forall(\mathsf{NE})$ *and* $\mathsf{RS}^\exists(\mathsf{Dom}) \Rightarrow \mathsf{RS}^\exists(\mathsf{SPE})$*.*

Proof. $\boxed{\mathsf{Win} \Rightarrow \mathsf{RS}^\forall(\gamma)}$ Let $\sigma_{\mathcal{P}}$ be a strategy profile such that for each player i, σ_i is winning for ϕ_i.

We first show that $\Sigma^\gamma_{\mathsf{G},\sigma_1}$ is not empty. For $\gamma \in \{\mathsf{NE}, \mathsf{SPE}\}$ this is because there always exist a subgame perfect equilibrium for Borel games (so in particular for Muller games) [32, Theorem 3.15] and a subgame perfect equilibrium is a Nash equilibrium. For $\gamma = \mathsf{Dom}$, note that by definition of dominant strategies, winning strategies are dominant, so $\Sigma^{\mathsf{Dom}}_{\mathsf{G},\sigma_1}$ contains at least σ_{-1}.

Let σ'_{-1} be a strategy profile for $\mathcal{P} \setminus \{1\}$. Since σ_1 is a winning we have that $\mathsf{G}, \sigma_1, \sigma'_{-1} \models \phi_1$. Therefore σ_1 is a solution for $\mathsf{RS}^\forall(\gamma)$.

$\boxed{\mathsf{RS}^\exists(\mathsf{Dom}) \Rightarrow \mathsf{RS}^\exists(\mathsf{SPE})}$ Let $\sigma_{\mathcal{P}}$ be a witness for $\mathsf{RS}^\exists(\mathsf{Dom})$. We define a strategy profile $\sigma'_{\mathcal{P}}$ such that σ'_i follows σ_i on all histories compatible with σ_i (that is if h prefix of $\rho \in \mathsf{Out}_\mathsf{G}(\sigma_i)$ then $\sigma'_i(h) = \sigma_i(h)$) and outside of these histories follows a subgame perfect equilibria: there always exist one for Borel games (so in particular for Muller games) [32, Theorem 3.15].

By definition of $\sigma'_{\mathcal{P}}$, the outcome $\mathsf{Out}_\mathsf{G}(\sigma'_{\mathcal{P}})$ is the same than $\mathsf{Out}_\mathsf{G}(\sigma_{\mathcal{P}})$. Because $\sigma_{\mathcal{P}}$ is a witness for $\mathsf{RS}^\forall(\mathsf{Dom})$, this outcome is winning for player 1.

It remains to show that σ'_{-1} is a subgame perfect equilibria. Let h be a history, i be a player different from player 1, and σ''_i be a strategy for player i. We show that from h player i does not improve by switching from σ'_i to another strategy σ''_i, which will show that $\sigma'_{\mathcal{P}} \circ h$ is a Nash equilibrium from h.

If h is compatible with σ_i then σ'_i coincide with σ_i from this history, so $\mathsf{Out}_\mathsf{G}(\sigma'_i, \sigma_{-i}) = \mathsf{Out}_\mathsf{G}(\sigma_{\mathcal{P}})$. Since σ_i is a dominant strategy, if $\mathsf{G}, \sigma''_i, \sigma_{-i} \models \phi_i$ then $\mathsf{G}, \sigma_i, \sigma_{-i} \models \phi_i$ and therefore this implies that $\mathsf{Out}_\mathsf{G}(\sigma'_i, \sigma_{-i})$ satisfy ϕ_i. This means that i does not improve by switching from σ'_i to σ''_i.

If h is not compatible with σ_i, then σ'_i plays according to a subgame-perfect equilibria since the first deviation. In particular, this strategy is a Nash equilibrium from h.

This shows that σ'_{-1} is a subgame perfect equilibrium and has $\sigma'_{\mathcal{P}} \models \phi_1$, this is a witness for $\mathsf{RS}^\exists(\mathsf{SPE})$.

$\boxed{\mathsf{RS}^{\forall}(\mathsf{Dom}) \not\Rightarrow \mathsf{RS}^{\forall}(\mathsf{NE})}$ Consider the example given in Figure 6. The strategy r for player 2 is dominant and any strategy of player 3 is dominant. The outcome of these strategies always go to the bottom state where ϕ_{sys} is satisfied. Therefore there is a solution to $\mathsf{RS}^{\forall}(\mathsf{Dom})$.

However, we show that there is no solution to $\mathsf{RS}^{\forall}(\mathsf{NE})$. Consider the strategy profile $(\cdot, l, b)$, this is a Nash equilibrium (even a subgame Nash equilibrium) since no player can improve his/her strategy. Note that player 1 is losing for that profile, hence no strategy of player 1 can ensure that it will win for all Nash equilibria. $\square$

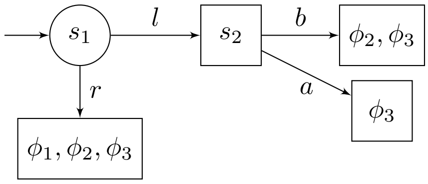

Figure 6: Example showing that $\mathsf{RS}^{\forall}(\mathsf{Dom}) \not\Rightarrow \mathsf{RS}^{\forall}(\mathsf{NE})$. Player 2 controls circle states, player 3 square states and player 1 does not control any state.

In the example of Section 4, we saw that more strategy profiles satisfied the assume-guarantee condition compared to assume-admissibility, including undesirable strategy profiles. We show that the rule $\mathsf{AG}^{\wedge}$ is indeed more often satisfied than AA; while the rules $\mathsf{AG}^{\vee}$, and AA are incomparable.

Lemma 26. *We have* $\mathsf{AG}^{\wedge} \not\Rightarrow \mathsf{AA}$*;* $\mathsf{AG}^{\vee} \not\Rightarrow \mathsf{AA}$*;* $\mathsf{AA} \not\Rightarrow \mathsf{AG}^{\wedge}$ *and* $\mathsf{AA} \not\Rightarrow \mathsf{AG}^{\vee}$.

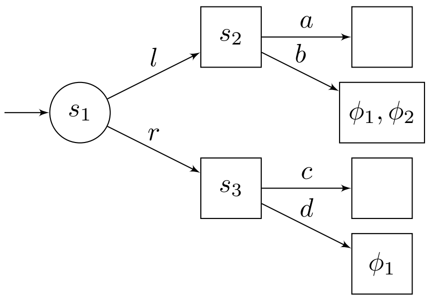

Figure 7: Example showing that $\mathsf{AG} \not\Rightarrow \mathsf{AA}$. Player 1 controls circle states and player 2 square states.

Proof. $\boxed{\mathsf{AG}^{\wedge} \not\Rightarrow \mathsf{AA} \text{ and } \mathsf{AG}^{\vee} \not\Rightarrow \mathsf{AA}}$ Consider the game represented in Figure 7. In this example, we have $\mathsf{Adm}_1 = \Sigma_1$. Therefore player 2 has no winning strategy against all admissible strategies of Adm_2 (in particular the strategy of player 1 that plays r, makes player 2 lose). So AA fails. However, we do have $\mathsf{AG}^{\wedge}$ by the profile $\sigma_1\colon s_1 \mapsto l, \sigma_2\colon s_2 \mapsto b, s_3 \mapsto c$. This profile also satisfies $\mathsf{AG}^{\vee}$ which is equivalent to $\mathsf{AG}^{\wedge}$ for two player games.

$\boxed{\mathsf{AA} \not\Rightarrow \mathsf{AG}^{\wedge}}$ Consider the example of Figure 8. The profile where player 1 and player 2 plays to the right is assume-admissible. However there is no solution to assume-guarantee synthesis: if player 1 and player 2 change their strategies to go to the state labeled ϕ_1, ϕ_2, then the condition $\mathcal{G}, \sigma_3 \models (\phi_1 \wedge \phi_2) \Rightarrow \phi_3$ is not satisfied.

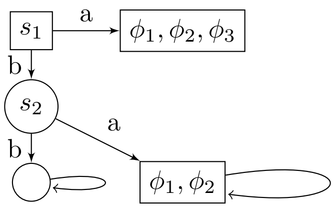

Figure 8: Example showing that $\mathsf{AA} \not\Rightarrow \mathsf{AG}^{\wedge}$. Player 1 controls circle states and player 2 square states; player 3 does not control any state.

$\boxed{\mathsf{AA} \not\Rightarrow \mathsf{AG}^{\vee}}$ We will provide a counter-example to show our claim. Note that we need strictly more than two players since otherwise $\mathsf{AG}^{\vee}$ is equivalent to $\mathsf{AG}^{\wedge}$, and we have just shown that AA implies $\mathsf{AG}^{\wedge}$.

Consider the game with three players in Fig. 9. Define the following objectives: $\phi_1 = \mathtt{GF}(s_4 \vee s_7)$, $\phi_2 = \mathtt{GF}(s_4 \vee s_6)$, $\phi_3 = \mathtt{true}$, where ϕ_i is player i's objective. These are actually reachability objectives since the game ends in absorbing states.

Now, action b is dominated at states s_2 and s_3 for player 2. Thus player 1 has a AA-winning strategy which consists in taking a at s_1. Player 2 has a winning strategy in the game (taking a at both states). Player 3 has a AA-winning strategy too since actions b are eliminated for player 2. Therefore, there is an AA-winning strategy profile which ends in s_4.

On the other hand, there is no $\mathsf{AG}^{\vee}$ profile. In fact, player 1 has no winning strategy to ensure $\phi_2 \vee \phi_3 \Rightarrow \phi_1$, which is equivalent to ϕ_1 since $\phi_3 = \mathtt{true}$.

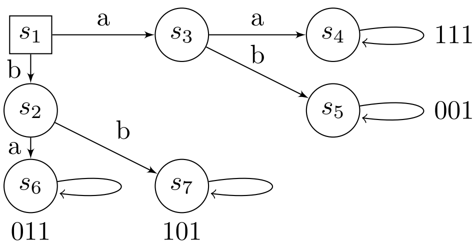

Figure 9: Example showing that $\mathsf{AA} \not\Rightarrow \mathsf{AG}^{\vee}$. Player 1 controls circle states and player 2 square states; player 3 does not control any state. At each absorbing state, the given Boolean vector represents the set of players for which the state is winning.

$\square$

Lemma 27. *For two player games,* $\mathsf{AA} \Rightarrow \mathsf{AG}^{\wedge}$*.*

Proof. Assume $\mathcal{G}$ is a two player game and consider strategy profile (σ_1, σ_2) witness of AA. Note that if player j decreases his own value at position k then its value for $h_{\leq k+1}$ will be smaller or equal to 0 which means player j has no winning strategy from this history. By determinacy of turn-based zero-sum games, player $3 - j$ has a winning strategy for $\neg\phi_j$. Therefore we can adjust the strategies (σ_1, σ_2) such that if there is a player j that decreases his own value, the other player will make it lose. We write (σ'_1, σ'_2) the strategies thus defined and we will show that they form a solution of Assume-Guarantee.

Let ρ be the outcome of the strategy profile (σ'_1, σ'_2). We can show that ρ is also the outcome of (σ_1, σ_2). First we recall that an admissible strategy does not decrease his own value (Lemma 4). Therefore each σ'_i is identical to σ_i on the run ρ. By Theorem 1, ρ satisfies $\phi_1 \wedge \phi_2$.

Let σ''_1 be an arbitrary strategy profile for 1, and consider $\rho' = \mathsf{Out}_{\mathsf{G}}(\sigma''_1, \sigma'_2)$. We show that $\rho' \models \phi_1 \Rightarrow \phi_2$. Note that player 2 cannot be the first to decrease its value during ρ' since it behave according to σ_2 has long has there are no devition, and σ_2 is admissible and admissible strategies do not decrease their own values.

- If player 1 decreases its value during ρ', player 2 will play to make him lose and $\rho' \not\models \phi_1$. As a consequence $\rho' \models \phi_1 \Rightarrow \phi_2$.
- Otherwise no player decreases his own value during ρ'. We assume that $\rho' \models \phi_1$ and show that $\rho' \models \phi_2$. Since $\rho' \models \phi_1$, by Lemma 8, there is a strategy τ''_1 which is admissible and compatible with ρ'. Since ρ' is an outcome of σ'_2, and of τ''_1, we have $\mathsf{Out}_{\mathsf{G}}(\tau''_1, \sigma_2) = \rho'$. Now, since τ''_1 is admissible and by the fact that σ_2 satisfies the condition 2 of AA, we obtain $\rho' \models \phi_2$, which proves the property.

We can show the same property replacing the roles of player 1 and player 2, thus showing that the profile is solution of $\mathsf{AG}^\wedge$. □

We now consider several non-implications of Figure 5.

Lemma 28. *$\mathsf{AA} \not\Rightarrow \mathsf{Win}$, $\mathsf{AG}^\wedge \not\Rightarrow \mathsf{Win}$, $\mathsf{AG}^\vee \not\Rightarrow \mathsf{Win}$, $\mathsf{Coop} \not\Rightarrow \mathsf{AA}$, $\mathsf{Coop} \not\Rightarrow \mathsf{AG}^\wedge$, $\mathsf{Coop} \not\Rightarrow \mathsf{RS}^\exists(\mathsf{Dom})$, and for all $\gamma \in \{\mathsf{NE}, \mathsf{SPE}, \mathsf{Dom}\}$, $\mathsf{RS}^{\exists,\forall}(\gamma) \not\Rightarrow \mathsf{Coop}$.*

Proof. $\boxed{\mathsf{AA} \not\Rightarrow \mathsf{Win}}$ Towards a contradiction assume $\mathsf{AA} \Rightarrow \mathsf{Win}$, then since we have $\mathsf{Win} \Rightarrow \mathsf{AG}^\wedge$ (Lemma 24), we would have $\mathsf{AA} \Rightarrow \mathsf{AG}^\wedge$ but this contradicts Lemma 26.

$\boxed{\mathsf{AG}^\wedge \not\Rightarrow \mathsf{Win}}$ By Lemma 27, we have $\mathsf{AA} \Rightarrow \mathsf{AG}^\wedge$, so $\mathsf{AG}^\wedge \Rightarrow \mathsf{Win}$ would imply, by transitivity, $\mathsf{AA} \Rightarrow \mathsf{Win}$, which contradicts the previous case.

$\boxed{\mathsf{AG}^\vee \not\Rightarrow \mathsf{Win}}$ Towards a contradiction assume $\mathsf{AG}^\vee \Rightarrow \mathsf{Win}$, then since we have $\mathsf{Win} \Rightarrow \mathsf{AA}$ (Lemma 24), we would have $\mathsf{AG}^\vee \Rightarrow \mathsf{AA}$ but this contradicts Lemma 26.

$\boxed{\mathsf{Coop} \not\Rightarrow \mathsf{AA}}$ In Figure 7, we have an example of a game where there is no solution for AA (see the proof of Lemma 26 for details), however there is a solution for Coop: (l, b).

$\boxed{\mathsf{Coop} \not\Rightarrow \mathsf{AG}^\wedge}$ Consider the example of Figure 10. There is a solution for Coop: player 1 plays a. However there is no solution for $\mathsf{AG}^\wedge$: player 2 has no strategy to ensure that $\phi_1 \implies \phi_2$.

$\boxed{\mathsf{Coop} \not\Rightarrow \mathsf{RS}^\exists(\mathsf{Dom})}$ Consider the example of Figure 11. This example has a solution for Coop, for instance (l, ac) or (r, bd). However player 2 has no dominant strategy: l looses against bd so it is dominated by r, and r looses against ac so it is dominated by l. Therefore $\mathsf{RS}^\exists(\mathsf{Dom})$ has no solution.

$\boxed{\mathsf{RS}^{\exists,\forall}(\gamma) \not\Rightarrow \mathsf{Coop}}$ Consider the example of Figure 12. There is no solution for Coop: player 2 can never win. However there is a solution for any concept in $\mathsf{RS}^{\exists,\forall}(\gamma)$: player 1 wins against any of the strategy satisfying these concepts since the only possible outcome is winning for him. □

In the controller synthesis framework using two-player games between a controller and its environment, some works advocate the use of environment objectives which the environment can guarantee against any controller [9]. Under this assumption, $\mathsf{Win\text{-}under\text{-}Hyp}$ implies AA:

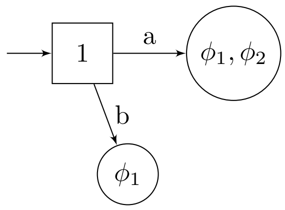

Figure 10: Example showing that $\mathsf{Coop} \not\Rightarrow \mathsf{AG}^{\wedge}$. Player 1 controls the circle state.

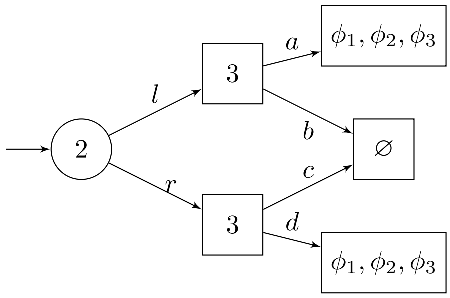

Figure 11: Example showing that $\mathsf{Coop} \not\Rightarrow \mathsf{RS}^{\exists}(\mathsf{Dom})$. Player 2 controls circle states, player 3 square states and player 1 does not control any state.

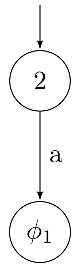

Figure 12: Two-player game showing that $\mathsf{RS}^{\exists,\forall}(\gamma) \not\Rightarrow \mathsf{Coop}$. Player 2 controls the circle state but has no choice.

Lemma 29. *Let $\mathsf{G} = \langle \mathsf{A}, \phi_1, \phi_2 \rangle$ be a two-player game. If player 2 has a winning strategy for ϕ_2 and* Win-under-Hyp *has a solution, then* AA *has a solution.*

Proof. Assume that σ_2^w is a winning strategy for ϕ_2 and let σ_1, σ_2 be a solution of Win-under-Hyp. We have that $\forall \sigma_2'.\ \sigma_1, \sigma_2' \models \phi_2 \Rightarrow \phi_1$ and $\forall \sigma_1'.\ \sigma_1', \sigma_2 \models \phi_1 \Rightarrow \phi_2$. Since σ_2^w is a winning strategy, all admissible strategies of player 2 are winning. Then, for all $\sigma_2' \in \mathsf{Adm}_2$, we have $\mathsf{G}, \sigma_1, \sigma_2' \models \phi_2$ and because $\forall \sigma_2', \sigma_1, \sigma_2' \models \phi_2 \Rightarrow \phi_1$, we also have that $G, \sigma_1, \sigma_2' \models \phi_1$. If σ_1 is dominated, there exists a non-dominated strategy σ_1^a that dominates it [3, Thm. 11], otherwise we take $\sigma_1^a = \sigma_1$. In both cases σ_1^a is admissible. As σ_1 is dominated by σ_1^a, $\mathsf{G}, \sigma_1, \sigma_2' \models \phi_1$ implies $\mathsf{G}, \sigma_1^a, \sigma_2' \models \phi_1$. This shows that the condition $\forall \sigma_2' \in \mathsf{Adm}_2(\mathsf{G}).\ \mathsf{G}, \sigma_1^a, \sigma_2' \models \phi_1$ is satisfied. Since σ_2^w is winning, it is admissible and we also have $\forall \sigma_1' \in \mathsf{Adm}_1(\mathsf{G}).\ \mathsf{G}, \sigma_1', \sigma_2^w \models \phi_2$. Therefore all conditions of Assume-Admissible are satisfied by (σ_1^a, σ_2^w).

□

Rectangularity We now consider the *robustness* of the profiles synthesized using the above rules. An AA-winning strategy profile $\sigma_{\mathcal{P}}$ is robust in the following sense: The set of AA-winning profiles is *rectangular*, *i.e. any* combination of AA-winning strategies independently chosen for each player, is an AA-winning profile. Second, if one replaces *any* subset of strategies in AA-winning profile $\sigma_{\mathcal{P}}$ by arbitrary admissible strategies, the objectives of all the other players still hold. Formally, a *rectangular set* of strategy profiles is a set that is a Cartesian product of sets of strategies, given for each player. A synthesis rule is *rectangular* if the set of strategy profiles satisfying the rule is rectangular. The RS rules require a specific definition since player 1 has a particular role: we say that $\mathsf{RS}^{\forall,\exists}(\gamma)$ is rectangular if for any strategy σ_1 witnessing the rule, the set of strategy profiles $(\sigma_2, \ldots, \sigma_n) \in \Sigma^{\gamma}_{\mathsf{G},\sigma_1}$ s.t. $\mathsf{G}, \sigma_1, \ldots, \sigma_n \models \phi_1$ is rectangular. We show that apart from AA, only Win and $\mathsf{RS}^{\forall}(\mathsf{Dom})$ are rectangular.

Theorem 8. *We have 1. Rule* AA *is rectangular; and for all games* G*,* AA*-winning strategy profile σ_P, coalition $C \subseteq \mathcal{P}$, if $\sigma_C' \in \mathsf{Adm}_C(\mathsf{G})$, then $\mathsf{G}, \sigma_{-C}, \sigma_C' \models \bigwedge_{i \in -C} \phi_i$. 2. The rules* Win *and* $\mathsf{RS}^{\forall}(\mathsf{Dom})$ *are rectangular; the rules* Coop*,* $\mathsf{AG}^{\vee}$*,* $\mathsf{AG}^{\wedge}$*,* $\mathsf{RS}^{\exists}(\mathsf{NE}, \mathsf{SPE}, \mathsf{Dom})$*, and* $\mathsf{RS}^{\forall}(\mathsf{NE}, \mathsf{SPE})$ *are not rectangular.*

Proof. $\boxed{\text{AA is rectangular}}$ If there is no solution to AA, then the set of witness is empty, and therefore is rectangular. If there is only one solution, then it is the Cartesian product of singletons and therefore also a rectangular set.

Otherwise let $\sigma_{\mathcal{P}}$ and $\sigma'_{\mathcal{P}}$ be two solutions of AA. Let i be a player of $\mathcal{P}$, we show that σ_i, σ'_{-i} is also a solution of AA. We have that $\sigma_i \in \mathsf{Adm}(\mathsf{G})$ and for all $j \neq i$, $\sigma_j \in \mathsf{Adm}(\mathsf{G})$, because condition 1 holds for $\sigma_{\mathcal{P}}$ and $\sigma'_{\mathcal{P}}$. Therefore condition 1 holds for σ_i, σ'_{-i}. Similarly, $\forall \sigma'_{-i} \in \mathsf{Adm}_{-i}(\mathsf{G})$. $\mathsf{G}, \sigma'_i, \sigma_i \models \phi_i$ and for all $j \neq i$, $\forall \sigma'_{-j} \in \mathsf{Adm}_{-j}(\mathsf{G})$. $\mathsf{G}, \sigma'_j, \sigma_j \models \phi_j$, because condition 2 holds for $\sigma_{\mathcal{P}}$ and $\sigma'_{\mathcal{P}}$. Therefore condition 2 holds for σ_i, σ'_{-i} and it is a witness of AA.

Let Σ_i^{aa} be the set of strategy σ_i such that there exists σ_{-i} such that σ_i, σ_{-i} is a witness of AA. We can show that the set of witness of AA is the Cartesian product of the Σ_i^{aa}. We obviously have that the set of solutions is included in $\prod_{i\in\mathcal{P}} \Sigma_i^{aa}$. Let $\sigma_{\mathcal{P}}$ be a profile in $\prod_{i\in\mathcal{P}} \Sigma_i^{aa}$, and $\sigma'_{\mathcal{P}}$ a witness of AA. We can replace for one i at a time, the strategy σ'_i by σ_i in $\sigma'_{\mathcal{P}}$ and by the small property we previously proved, the strategy profile stays a solution of AA. Therefore $\sigma_{\mathcal{P}}$ is a solution of AA. This shows that the set of solutions is the rectangular set $\prod_{i\in\mathcal{P}} \Sigma_i^{aa}$.

$\boxed{\sigma'_C \in \mathsf{Adm}_C(\mathsf{G}) \Rightarrow \mathsf{G}, \sigma_{-C}, \sigma'_C \models \bigwedge_{i\in -C} \phi_i}$ This claim follows from the definition of AA-winning strategy profiles, since each strategy is winning against admissible strategies.

Now Consider any game G and fix a profile $\sigma_{\mathcal{P}}$ such that $\mathsf{G}, \sigma_{\mathcal{P}} \models \bigwedge_{1\leq i\leq n} \phi_i$.

$\boxed{\mathsf{Win}}$ Assume $\sigma_{\mathcal{P}}$ is solution to Win, then each σ_i is a winning strategy. Let σ'_i be a strategy part of another profile solution to Win. Then the strategy σ'_i ensures ϕ_i against any strategy profile for $-i$. If we replace σ_i by σ'_i in the profile $\sigma_{\mathcal{P}}$ then the condition for Win are still satisfied. Thus the rule Win is rectangular.

$\boxed{\mathsf{RS}^{\forall}(\mathsf{Dom})}$ Let σ_1 be a solution of $\mathsf{RS}^{\forall}(\mathsf{Dom})$. Let $\sigma_2, \dots, \sigma_n$ and $\sigma'_2, \dots, \sigma'_n$ be profiles of $\Sigma^{\mathsf{Dom}}_{\mathsf{G},\sigma_1}$ such that $\sigma_1, \sigma_2, \dots, \sigma_n \models \phi_1$ and $\sigma_1, \sigma'_2, \dots, \sigma'_n \models \phi_1$. If we define a profile $\tau_2, \dots, \tau_n$ where each τ_i is either σ_i or σ'_i, then as each τ_i is dominant, we have $\sigma_1, \tau_2, \dots, \tau_n \models \phi_1$ because σ_1 is a solution of $\mathsf{RS}^{\forall}(\mathsf{Dom})$. Therefore the profile belongs to $\Sigma^{\mathsf{Dom}}_{\mathsf{G},\sigma_1}$ and makes σ_1 win. This shows that the rule is rectangular.

$\boxed{\mathsf{RS}^{\exists}(\mathsf{Dom})}$ Consider the example of Figure 13. Since player 2 and player 3 are always winning, all their strategies are dominant. There is only one strategy σ_1 for player 1 since it controls no state. The profiles (a, c) and (b, d) are strategies of $\Sigma^{\mathsf{Dom}}_{\mathsf{G},\sigma_1}$ such that σ_1 wins for ϕ_1, but the profile (σ_1, a, d) does not make ϕ_1 hold. The rule is therefore not rectangular.

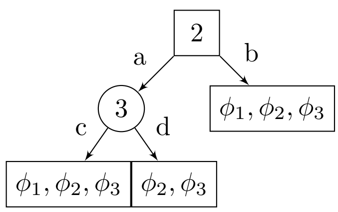

Figure 13: Game with three players showing that rule $\mathsf{RS}^{\exists}(\mathsf{Dom})$ is not rectangular. Here, player 1 controls no state; player 2 controls the square state, and player 3 controls the round state.

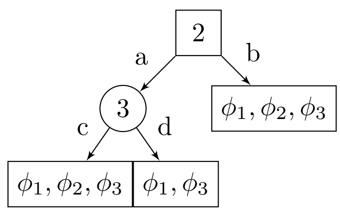

Figure 14: Game with three players showing that rule $\mathsf{RS}^{\forall}(\mathsf{NE})$ and $\mathsf{RS}^{\forall}(\mathsf{SPE})$ are not rectangular. Player 1 controls no state, player 2 controls the square state and player 3 the round state.

$\boxed{\mathsf{RS}^{\forall}(\mathsf{NE}) \text{ and } \mathsf{RS}^{\forall}(\mathsf{SPE})}$ Consider the game represented in Figure 14. Player 1 has only one strategy σ_1 and the other players have two possible strategies: a and b for player 2 and c and d for

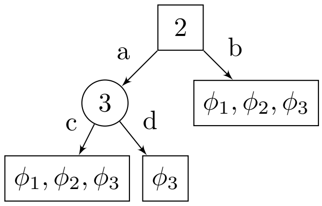

Figure 15: Game with three players showing that rule $\mathsf{RS}^{\exists}(\mathsf{NE})$ and $\mathsf{RS}^{\exists}(\mathsf{SPE})$ are not rectangular. Player 1 controls no state, player 2 controls the square state and player 3 the round state.

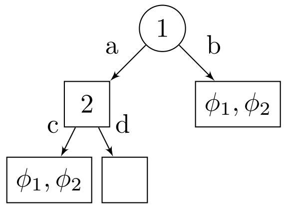

Figure 16: Game with two players showing that rule $\mathsf{AG}^{\vee}$ and $\mathsf{AG}^{\wedge}$ are not rectangular. Player 1 controls the round state and player 2 the square state.

player 3. Since player 1 is always winning, σ_1 is a solution for $\mathsf{RS}^{\forall}(\mathsf{NE}, \mathsf{SPE})$. The profiles (a, c) and (b, d) are two (subgame perfect) Nash equilibria which make ϕ_1 hold. However the profile (a, d) obtained by picking one strategy in each profile, is no longer a Nash equilibrium (and so not a subgame perfect equilibrium). Therefore $\mathsf{RS}^{\forall}(\mathsf{NE})$ and $\mathsf{RS}^{\forall}(\mathsf{SPE})$ are not rectangular.

$\boxed{\mathsf{RS}^{\exists}(\mathsf{NE}) \text{ and } \mathsf{RS}^{\exists}(\mathsf{SPE})}$ Consider the game represented in Figure 15. Player 1 has only one strategy and the other players have two possible strategies: a and b for player 2 and c and d for player 3. The profiles (a, c) and (b, d) are two (subgame perfect) Nash equilibria which make ϕ_1 hold. However the profile (a, d) obtained by taking one strategy in each profile, is no longer winning for player 1. Therefore $\mathsf{RS}^{\exists}(\mathsf{NE})$ and $\mathsf{RS}^{\exists}(\mathsf{SPE})$ are not rectangular.

$\boxed{\mathsf{Coop}}$ Once again, consider the game represented in Figure 15. The profiles (a, c) and (b, d) make all players win, but the profile (a, d), is no longer winning for the player 1, so it is not a solution of Coop. Therefore Coop is not rectangular.

$\boxed{\mathsf{AG}^{\vee} \text{ and } \mathsf{AG}^{\wedge}}$ Consider the game represented in Figure 16. The profiles (a, c) and (b, d) make the two players win. Since all possible outcome of the game satisfy the implications $\phi_1 \Rightarrow \phi_2$ and $\phi_2 \Rightarrow \phi_1$, both profiles are solution to $\mathsf{AG}^{\vee}$ and $\mathsf{AG}^{\wedge}$ (note that the two concepts coincide here because there are only two players). However the profile (a, d) obtained by taking one strategy in each profile, is no longer winning for the player 1. Therefore $\mathsf{AG}^{\vee}$ and $\mathsf{AG}^{\wedge}$ are not rectangular.

□

9 Conclusion

In this paper, we have introduced a novel synthesis rule, called the *assume admissible synthesis*, for the synthesis of strategies in non-zero sum n players games played on graphs with omega-regular objectives. We use the notion of admissible strategy, a classical concept from game theory, to take into account the objectives of the other players when looking for winning strategy of one player. We have compared our approach with other approaches such as assume guarantee synthesis and rational synthesis that target the similar scientific objectives. We have developed worst-case optimal algorithms to handle our synthesis rule as well as dedicated abstraction techniques.

The assume-admissible rule is useful to synthesize meaningful strategies which correctly take other players' expected behaviors into account. Nevertheless, the rule might suffer some limitations that we describe here. First, the restriction to admissible strategies can be questionable in some settings. This assumption is justified when the underlying agents are unknown but can be assumed to act rationally in the sense of admissibility, or simply when we want to actually synthesize a strategy profile and commit to using the AA rule during the whole process. The AA rule cannot be used, for instance, if the behaviors of some agents cannot be determined yet and cannot be assumed to be rational (in the sense of admissibility) either. Another issue is that the rule provides solutions less often than the cooperative synthesis rule in general, and the assume-guarantee rule for the case of two players (see Sect. 8). Hence, the rule might fail to find a solution even though there exists an appropriate strategy profile. A related observation is that since the rule assumes that each agent acts admissibly, the rule might yield sub-optimal solutions if an additional global criterion was given. Indeed, if we were to extend our synthesis problem by adding, say, a global quantitative optimization objective, then restricting to admissible strategies would mean to be sub-optimal in general, while the cooperative synthesis rule can give the optimal solution.

We have seen in Section 8 (Theorem 7) that a set of objectives $(\phi_1, \phi_2, \ldots, \phi_n)$ not having a solution for the AA rule can still have a solution with the Coop rule or with the $\mathsf{AG}^{\wedge}$ rule (for two players). Indeed, because the AA rule leads to solution spaces that are rectangular (Theorem 8), for a AA solution to exist, this requires the objectives to be strong enough so that strategies for each player can be determined compositionally. So, the solution cannot not rely on the synchronization of all the players on particular strategies. Nevertheless, if there exists a Coop solution for objectives $(\phi_1, \ldots, \phi_n)$, then there always exists a way to reinforce these original objectives so that there exists an AA solution. Indeed, if the regular play $w_1 \cdot (w_2)^\omega$ is a solution for Coop, then the stronger objectives $(\{w_1 \cdot (w_2)^\omega\}, \ldots, \{w_1 \cdot (w_2)^\omega\})$ has trivially a solution for the AA rule. As a future work, we will study the problem of reinforcing automatically a specification $(\phi_1, \phi_2, \ldots, \phi_n)$ that has no AA rule solution into a new specification $(\phi'_1, \phi'_2, \ldots, \phi'_n)$ which has a AA solution, while $(\phi'_1, \ldots, \phi'_n)$ is as weak as possible.

As further future work, we plan to investigate the admissibility notions on quantitative games, and to develop a tool prototype to support our assume admissible synthesis rule.

References

[1] Brandenburger Adam, Friedenberg Amanda, H Jerome, et al. Admissibility in games. *Econometrica*, 2008.

[2] Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *RAIRO-Theoretical Informatics and Applications-Informatique Théorique et Applications*, 36(3):261–275, 2002.

[3] Dietmar Berwanger. Admissibility in infinite games. In *Proc. of STACS'07*, volume 4393 of *LNCS*, pages 188–199. Springer, February 2007.

[4] Roderick Bloem, Rüdiger Ehlers, Swen Jacobs, and Robert Könighofer. How to handle assumptions in synthesis. In *SYNT'14*, volume 157 of *EPTCS*, pages 34–50, 2014.

[5] Romain Brenguier, Jean-François Raskin, and Ocan Sankur. Assume-Admissible Synthesis. In Luca Aceto and David de Frutos Escrig, editors, *26th International Conference on Concurrency Theory (CONCUR 2015)*, volume 42 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 100–113, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[6] Romain Brenguier, Jean-François Raskin, and Mathieu Sassolas. The complexity of admissibility in omega-regular games. In *CSL-LICS '14, 2014*. ACM, 2014.

[7] Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, and Jean-François Raskin. Doomsday equilibria for omega-regular games. In *VMCAI'14*, volume 8318, pages 78–97. Springer, 2014.

[8] Krishnendu Chatterjee and Thomas A Henzinger. Assume-guarantee synthesis. In *TACAS'07*, volume 4424 of *LNCS*. Springer, 2007.

[9] Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. Environment assumptions for synthesis. In *CONCUR 2008*, volume 5201 of *LNCS*, pages 147–161. Springer, 2008.

[10] Krishnendu Chatterjee, Thomas A Henzinger, and Marcin Jurdziński. Games with secure equilibria. *Theoretical Computer Science*, 365(1):67–82, 2006.

[11] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.

[12] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logics of Programs*, volume 131 of *LNCS*, pages 52–71. Springer, 1981.

[13] Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL'77*. ACM, 1977.

[14] Werner Damm and Bernd Finkbeiner. Automatic compositional synthesis of distributed systems. In *FM 2014*, volume 8442 of *LNCS*, pages 179–193. Springer, 2014.

[15] Luca de Alfaro, Patrice Godefroid, and Radha Jagadeesan. Three-valued abstractions of games: Uncertainty, but with precision. In *LICS'04*. IEEE, 2004.

[16] E Allen Emerson. Temporal and modal logic. *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, 995:1072, 1990.

[17] Marco Faella. Admissible strategies in infinite games over graphs. In *MFCS 2009*, volume 5734 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2009.

[18] Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. In *TACAS'10*, volume 6015 of *LNCS*, pages 190–204. Springer, 2010.

[19] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991. Translated into Chinesse by Renin University Press, Bejing: China.

[20] Zhu Han. *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge University Press, 2012.

[21] Thomas A. Henzinger, Rupak Majumdar, Freddy Y. C. Mang, and Jean-François Raskin. Abstract interpretation of game properties. In *SAS*, pages 220–239, 2000.

[22] Paul Hunter. *Complexity and Infinite Games on Finite Graphs*. PhD thesis, Computer Laboratory, University of Cambridge, 2007.

[23] Paul Hunter and Anuj Dawar. Complexity bounds for regular games. In Joanna Jedrzejowicz and Andrzej Szepietowski, editors, *MFCS'05*, volume 3618 of *LNCS*. Springer, 2005.

[24] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. In *Proc. 12th European Conference on Multi-Agent Systems*, LNCS. Springer, 2014.

[25] David E Long, Anca Browne, Edmund M Clarke, Somesh Jha, and Wilfredo R Marrero. An improved algorithm for the evaluation of fixpoint expressions. In *CAV'94*, volume 818 of *LNCS*. Springer, 1994.

[26] Zohar Manna and Pierre Wolper. Synthesis of communicating processes from temporal logic specifications. In *Logics of Programs*, volume 131 of *LNCS*, pages 253–281. Springer, 1981.

[27] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. Reasoning about strategies. In *FSTTCS 2010*, volume 8 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

[28] John Nash. Equilibrium points in n-person games. *Proc. NAS*, 1950.

[29] Daniel Neider, Roman Rabinovich, and Martin Zimmermann. Down the borel hierarchy: Solving muller games via safety games. *Theoretical Computer Science*, 560:219–234, 2014.

[30] Amir Pnueli. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57. IEEE, 1977.

[31] Helmut Seidl. Fast and simple nested fixpoints. *Inf. Proc. Letters*, 59(6):303–308, 1996.

[32] Michael Ummels. *Stochastic multiplayer games: theory and algorithms.* Amsterdam University Press, 2010.

A Proofs of Section 4

Lemma 1. *Any strategy for* Controller *is admissible if, and only if it satisfies (C0), (C1), and (C2) at all histories.*

Proof. Consider a strategy which does not satisfy the conditions. Note that (C0) cannot be violated. So we consider the two remaining cases.

Consider strategy σ that violates (C1), say, at history h. We define $\sigma' = \sigma[h \leftarrow \sigma_C]$, and show that σ is dominated by σ'. In fact, both strategies are identical on outcomes that do not admit h as prefix. Now, given any strategy τ for Scheduler compatible with h, consider $\tau' = \tau[h \leftarrow \sigma_S]$. Clearly, all outcomes that extend h and compatible with σ are losing for Controller – since the safety specification fails after h, while the outcomes compatible with σ' and τ' is winning. This shows that σ' dominates σ.

If σ violates (C2), this means that at some history h where a_i just became pending, some history h' compatible with σ, and extending h for k steps, does not set r_i to true although r_i is not constantly pending.

Let h'' denote the longest prefix of h' that ends in a Controller state. By assumption r_i is not pending at h'' and $\sigma(h'')$ does not set r_i to true. We define σ' identical to σ for all histories that do not admit h'' as prefix. From history h'', σ' sets r_i to true, and then sets r_{3-i} to true in the next round, and then always plays $\perp$. Consider strategies for User and Scheduler that are compatible with h'', and from h'', constantly play $\perp$ and q_1q_2 respectively (they are defined arbitrarily elsewhere). It follows that the generated outcome compatible with σ' extending h'' is winning for Controller; while all outcomes of σ extending h'' are losing since they immediately violate $\phi_{\mathsf{Controller}}$. Since all other outcomes of σ' which do not extend h'' are identical to those of σ, this shows that σ is dominated by σ'.

Conversely, consider any strategy σ that satisfies (C0)–(C2), and assume, towards a contradiction that there exists σ' that dominates σ. Consider any finite history h such that $\sigma(h) \neq \sigma'(h)$, and such that σ' has an outcome that extends h and satisfies $\phi_{\mathsf{Controller}}$. Such a history exists since there is a strategy profile for other players against which σ' wins but not σ. Note that $\phi_{\mathsf{Controller}}$ is not violated by h since σ' has a compatible outcome extending h that satisfies this property.

We construct an outcome ρ compatible with σ that extends h and satisfies $\phi_{\mathsf{Controller}}$ as follows. We set both a_1, a_2 constantly to false, and q_1, q_2 to true from history $h\sigma(h)$, while the variables of Controller are chosen according to σ. Thus, no request is pending when Scheduler plays at history $h\sigma(h)$. By (C2), the outcome satisfies $\mathtt{G}(\forall i, a_i \rightarrow \mathtt{F}_{\leq k} r_i)$. In fact, we know that the property is satisfiable from h (since σ' satisfies it), and none of the requests are pending after $h\sigma(h)$. Moreover, by (C1), we also have the second part of the formula. Hence, the outcome satisfies $\phi_{\mathsf{Controller}}$.

We now prove that σ' cannot dominate σ. In fact, let (τ_U, τ_S) be a strategy profile compatible with ρ, which at history $h\sigma'(h)$ switches to $(\sigma_U, \hat{\sigma}_S)$. It follows that $(\sigma, \tau_U, \tau_S) \models \phi_{\mathsf{Controller}}$ while $(\sigma', \tau_U, \tau_S) \not\models \phi_{\mathsf{Controller}}$, a contradiction. ☐

Lemma 2. *Any* Scheduler *strategy is admissible if, and only if it satisfies (C3), (C4), (C5), and (C6) at all histories.*

Proof. Consider a strategy τ that does not satisfy one of the conditions at history h. Then, all outcomes that extend h are losing for Scheduler since the temporal constraints on both requests cannot be satisfied. We describe a strategy τ' that dominates τ. Assume that τ violates (C3) at h. We define τ' from τ, which is identical to τ at all histories that do not contain h as prefix. At h, τ' sets q_1 to true in the first round, and q_2 in the second round, and continues as σ_S. Now, for any strategy of Controller which, at h, plays $\perp$ twice, and switches to σ_C, all outcomes from h compatible with both described strategies satisfy $\phi_{\mathsf{Scheduler}}$. This shows that τ' dominates τ. The case of other conditions (C4) or (C5) being violated by τ are treated similarly. Note that (C6) cannot be violated by definition.

Conversely, we show that any strategy τ that satisfy (C3)–(C6) is admissible. Consider any admissible strategy τ' that dominates τ; we will show a contradiction. Note that τ' must satisfy all four conditions by the previous case. Let h be a maximal finite prefix compatible with τ and τ' with $\tau(h) \neq \tau'(h)$, and such that some outcome compatible with τ' extending h is winning for Scheduler. Such a history exists since the two strategies must be different, and because τ' dominates τ.

Define a Controller strategy σ compatible with h, which

- from $h\tau(h)$, constantly sets all r_i to false,
- and from $h\tau'(h)$, constantly sets all r_i to true.

Note that all outcomes are losing from $h\tau'(h)$ (this behavior corresponds to $\hat{\sigma}_C$). It thus suffices to show that some outcome compatible with τ and σ, and extending $h\tau(h)$ is winning for Scheduler to obtain a contradiction.

First, observe that no prefix of h satisfies (C6) since from the same history, τ' has a winning possible outcome. In particular, this is the case of h itself. Since $\tau(h) \neq \tau'(h)$ although both satisfy (C3)–(C5), h does not satisfy any of the hypotheses of these conditions. It must be that no request is pending from the previous round (that is, in the previous round, r_2 was false, and either r_1 was false or it was followed by q_1). So in the current round, either no request was made, or only one request was made. It follows that any strategy satisfying (C3)–(C5) sets q_1 and q_2 so as to satisfy $\phi_{\mathsf{Scheduler}}$, given that no new request is made under σ. Thus, this particular outcome is compatible with τ and satisfies $\phi_{\mathsf{Scheduler}}$, contradicting that τ' dominates τ. ☐

Lemma 3. *For all $k \geq 4$, all strategy profiles $(\sigma_U, \sigma_C, \sigma_S)$ satisfying (C1)–(C6) also satisfy $\phi_{\mathsf{User}} \wedge \phi_{\mathsf{Controller}} \wedge \phi_{\mathsf{Scheduler}}$.*

Proof. We first show that $\phi_{\mathsf{Scheduler}}$ holds for all outcomes under condition (C1) and (C3)–(C5). Let us denote by (H3), (H4), and (H5) the hypotheses of conditions (C3), (C4), and (C5) respectively. We also define the following:

(H6) No request is pending from the previous round, and at most one request is made in the current round.

We are going to show that under these conditions, any history ending at Scheduler's states satisfy one of the conditions among (H3)–(H6). We proceed by induction on the length of the history.

Initially, this is the case since if both requests are made by Controller, then we are in (H3). Otherwise, at most one request is made, and (H6) holds.

Assume now that this holds in the previous round. If (H3) holds, then by (C3), (H4) holds in the next round. If (H4) holds, then by (C4), either (H5) or (H6) hold in the next round. If (H5) holds, then by (C5), either (H4) or (H6) holds in the next round. If (H6) holds, and no request is pending, then (H6) holds in the next round. If some request is pending, by (C1), we are in (H3) or (H4) in the next round depending on which request is pending.

Since in each case the corresponding request is granted in time, this shows that any outcome satisfying these conditions satisfies $\phi_{\mathsf{Scheduler}}$.

Let us argue that $\phi_{\mathsf{Controller}}$ holds too. First, (C1) implies $\mathtt{G}(\forall i, r_i \rightarrow X(\neg r_i \mathtt{W} q_i))$ by definition. Second, by $\phi_{\mathsf{Scheduler}}$, all requests are granted in at most two rounds, thus for any incoming action a_i, if r_i is pending (that is, r_i was set to true in the previous round), q_i will be set to true in the current round; thus it will not be pending in the beginning of the next round (note that if r_i is set to true in the next round, it arrives precisely 4 steps after a_i). It follows from (C2) that for each pending action a_i, r_i is set to true within k steps. Thus $\phi_{\mathsf{Controller}}$ holds too. ☐

B Admissible Strategies not Monotonous under Game Abstractions

In [15], *existential* and *universal abstractions* are defined on two-player games, which generalize the corresponding abstraction technique for automata. Given a game G, the idea is to obtain a game $\underline{G}$ by applying state-space abstraction, which is harder to win, and conversely, a game $\overline{G}$ which is easier to win. Any winning strategy for $\underline{G}$ can be mapped to a winning strategy in G, and conversely, if $\overline{G}$ cannot be won, then nor can G be. The goal is to only explore the abstract state space to check a sufficient condition for the existence of a winning strategy, or to prove there is none.

More precisely, given a partition $S_1, \ldots, S_n$ of the states respecting the players, the game $\underline{G}$ is defined on states $\{S_1, \ldots, S_n\}$. If S_i is made of player 1 states, then we put an edge from S_i to S_j if from *all* $s \in S_i$ there is an edge to some state of S_j in G. We say that any edge from s to S_j is *mapped* to the edge (S_i, S_j). If S_i is made of player 2 states, then we put an edge (S_i, S_j) if for *some* state $s \in S_i$, there is an edge from s to S_j.

The game $\overline{G}$ is defined by inverting the abstractions applied to both players.

One can wonder whether admissible strategies of $\underline{G}$ map to admissible strategies of G, or whether there is any relation of this sort. We answer this question negatively in the following remark.

Remark 2. *Given a strategy $\underline{\sigma}$ of $\underline{G}$, we consider its concretization $\gamma(\underline{\sigma})$ as the set of strategies of G, which from any history h, take edges that map to the edge $\underline{\sigma}(\alpha h)$, where αh is the projection of h in the abstract state space.*

We give some examples to show that any of the following cases is possible:

- $\mathsf{Adm}_1(G) \subseteq \gamma(\mathsf{Adm}_1(\underline{G}))$,
- $\gamma(\mathsf{Adm}_1(\underline{G})) \subseteq \mathsf{Adm}_1(G)$,
- $\mathsf{Adm}_1(G) \not\subseteq \gamma(\mathsf{Adm}_1(\underline{G}))$, *and* $\gamma(\mathsf{Adm}_1(\underline{G})) \not\subseteq \mathsf{Adm}_1(G)$,

showing that $\underline{G}$ cannot be used to derive systematically an under- or over-approximation of the set $\mathsf{Adm}_1(G)$.

Consider the game in Fig. 17, where Player 1 plays from circular states and has the safety objective of avoiding $\times$. The original game G is given on the left. On the right, $\underline{G}$ is given, which is obtained by existential abstraction by merging states $1, 2$.

We will refer to edges by their labels, and identify strategies with edges since there is a single state for Player 1. We have $\mathsf{Adm}_1(G) = \{a\}$ but $\gamma(\mathsf{Adm}_1(\underline{G})) = \{a, b\}$. In fact, The only edge from state 0 defines an admissible strategy (which is also the only strategy), and both edges a and b of G map to this edge. This shows $\mathsf{Adm}_1(G') \subseteq \gamma(\mathsf{Adm}_1(\underline{G'}))$.

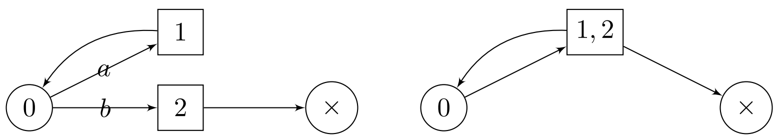

Figure 17: Game G and its abstraction $\underline{G}$.

As a second example, consider the following game where the goal is to reach the state ✓. The only admissible strategy in $\underline{G'}$ is taking the edge a, so $\gamma(\mathsf{Adm}_1(\underline{G'})) = \{a\}$. However, in the original game G', we have $\mathsf{Adm}_1(G') = \{a, b\}$ since both are winning. This shows $\gamma(\mathsf{Adm}_1(\underline{G'})) \subseteq \mathsf{Adm}_1(G')$.

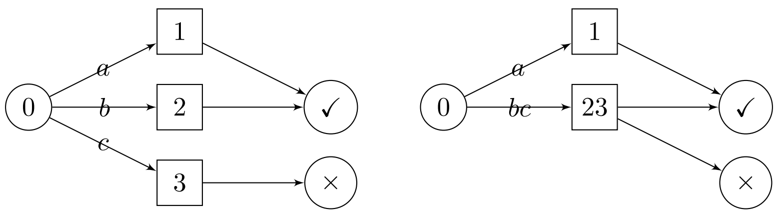

Figure 18: Game G' and its abstraction $\underline{G'}$.

More generally, by combining the two small games given here (for instance, taking their union), one can obtain games where $\mathsf{Adm}_1(G)$ and $\gamma(\mathsf{Adm}_1(\underline{G}))$ are incomparable.